

PATENT OFFICE  
JAPANESE GOVERNMENT

This is to certify that the annexed is a true copy of the following  
application as filed with this Office.

Date of Application: April 11, 2000

Application Number: Patent Application  
No. 2000-109416

Applicant(s): FUJITSU LIMITED

December 22, 2000

Commissioner,  
Patent Office Kozo Oikawa

Certificate No. 2000-3105862

日 本 国 特 許 庁  
PATENT OFFICE  
JAPANESE GOVERNMENT

J1046 U.S. PRO  
09/804268  
03/13/01

別紙添付の書類に記載されている事項は下記の出願書類に記載されて  
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed  
with this Office.

出 願 年 月 日  
Date of Application:

2000年 4月11日

出 願 番 号  
Application Number:

特願2000-109416

出 願 人  
Applicant (s):

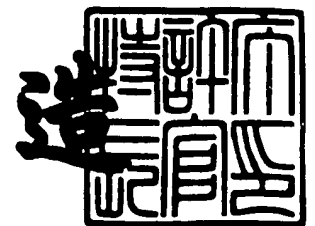
富士通株式会社

CERTIFIED COPY OF  
PRIORITY DOCUMENT

2000年12月22日

特 許 庁 長 官  
Commissioner,  
Patent Office

及 川 耕 造



出証番号 出証特2000-3105862

【書類名】 特許願

【整理番号】 0050603

【提出日】 平成12年 4月11日

【あて先】 特許庁長官殿

【国際特許分類】 G06G 9/06

【発明の名称】 G U I 系プログラムのテスト支援装置および支援方法

【請求項の数】 12

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

【氏名】 村石 誠

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

【氏名】 殿村 方規

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

【氏名】 藤川 泰之

【特許出願人】

【識別番号】 000005223

【氏名又は名称】 富士通株式会社

【代理人】

【識別番号】 100074099

【住所又は居所】 東京都千代田区二番町8番地20 二番町ビル3F

【弁理士】

【氏名又は名称】 大菅 義之

【電話番号】 03-3238-0031

【選任した代理人】

【識別番号】 100067987

【住所又は居所】 神奈川県横浜市鶴見区北寺尾 7 - 2 5 - 2 8 - 5 0 3

【弁理士】

【氏名又は名称】 久木元 彰

【電話番号】 045-573-3683

【手数料の表示】

【予納台帳番号】 012542

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9705047

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 G U I 系プログラムのテスト支援装置および支援方法

【特許請求の範囲】

【請求項 1】 グラフィックユーザインタフェースを使用する画面プログラムのテストを支援する装置において、

テスト対象画面プログラムの画面定義情報を入力として受け取り、該テスト対象画面プログラムのクラスに対してオブジェクト指向の継承の関係におけるサブクラスであって、画面プログラムをテストするためのテスト支援クラスを生成するテスト支援クラス生成手段と、

該生成されたテスト支援クラスを使用して、テスト対象画面プログラムのテストを実行するテスト実行手段とを備えることを特徴とする G U I 系プログラムのテスト支援装置。

【請求項 2】 前記テスト支援装置において、

前記画面定義情報から前記テスト対象画面プログラムに対するテスト仕様を生成して前記テスト実行手段に与えるテスト仕様生成手段を更に備えることを特徴とする請求項 1 記載の G U I 系プログラムのテスト支援装置。

【請求項 3】 前記テスト支援装置において、

前記テスト仕様生成手段によって生成されたテスト仕様と、前記テスト実行手段によるテスト実行結果とを用いてテスト報告を生成するテスト報告生成手段を更に備えることを特徴とする請求項 2 記載の G U I 系プログラムのテスト支援装置。

【請求項 4】 前記テスト支援クラスが入力テストデータの入力支援を行う機能を備えることを特徴とする請求項 1、2、または 3 記載の G U I 系プログラムのテスト支援装置。

【請求項 5】 前記テスト支援クラスがテストの手動、または自動実行時のテスト結果を記録する機能を備えることを特徴とする請求項 1、2、または 3 記載の G U I 系プログラムのテスト支援装置。

【請求項 6】 前記テスト支援クラスが画面上でのテスト実行箇所の視覚的表示を行う機能を備えることを特徴とする請求項 1、2、または 3 記載の G U I

系プログラムのテスト支援装置。

【請求項 7】 前記テスト支援クラスが新たな入力テストデータ、または以前のテスト実行結果に対する入力データを用いたテストの手動、または自動実行の機能を備えることを特徴とする請求項 1、2、または 3 記載の G U I 系プログラムのテスト支援装置。

【請求項 8】 前記テスト支援クラスが前記以前のテスト実行結果に対する入力データを用いたテストの実行結果が該以前の実行結果と異なる時に警告の表示を行う機能を更に備えることを特徴とする請求項 7 記載の G U I 系プログラムのテスト支援装置。

【請求項 9】 前記テスト支援クラスが前記テスト対象画面プログラムの実行性能の測定を支援する機能を備えることを特徴とする請求項 1、2、または 3 記載の G U I 系プログラムのテスト支援装置。

【請求項 1 0】 グラフィックユーザインタフェースを仕様する画面プログラムのテストを支援する方法において、

テスト対象画面プログラムの画面定義情報を入力として受け取り、該テスト対象画面プログラムのクラスに対してオブジェクト指向の継承の関係におけるサブクラスであって、画面プログラムをテストするためのテスト支援クラスを生成し

該生成されたテスト支援クラスを使用してテスト対象画面プログラムのテストを実行することを特徴とする G U I 系プログラムのテスト支援方法。

【請求項 1 1】 グラフィックユーザインタフェースを使用する画面プログラムのテストを支援するための計算機で使用される記憶媒体において、

テスト対象画面プログラムの画面定義情報を入力として受け取るステップと、  
該テスト対象画面プログラムのクラスに対してオブジェクト指向の継承の関係におけるサブクラスであって、画面プログラムをテストするためのテスト支援クラスを生成するステップとを計算機に実行させるためのプログラムを格納した計算機読み出し可能可搬型記憶媒体。

【請求項 1 2】 グラフィックユーザインタフェースを使用する画面プログラムのテストを実行するための計算機で使用される記憶媒体において、

新たな入力テストデータ、または以前のテスト実行結果に対する入力データを  
受け取るステップと、

該受け取った入力データを用いて、テスト対象画面プログラムのクラスに対し  
てオブジェクト指向の継承の関係におけるサブクラスであって、画面プログラム  
をテストするためのテスト支援クラスを使用して該テスト対象画面プログラムの  
テストを実行するステップとを計算機に実行させるためのプログラムを格納した  
計算機読み出し可能可搬型記憶媒体。

【発明の詳細な説明】

【 0 0 0 1 】

【発明の属する技術分野】

本発明はプログラムのテスト方式に係り、更に詳しくはグラフィックユーザイ  
ンタフェースを使用する画面プログラムのテスト作業の効率化、および自動化を  
実現するための G U I 系プログラムのテスト支援装置、および支援方法に関する

【 0 0 0 2 】

【従来の技術と発明が解決しようとする課題】

グラフィックユーザインタフェース（G U I）を使用する画面プログラムに対  
するテストを行う場合には、一般的なプログラムのテストに用いられるようなテ  
スト用ドライバを作成することは困難であり、主にプログラム作成者による手作  
業でテストが行われていた。一般にキーボードやマウスの操作をスクリプト化（  
文字記述化）するようなテスト支援ツールは存在するが、そのようなテスト支援  
ツールは画面の位置や画面上の部品の配置が変わって使えなくなるというような  
不便な点があった。

【 0 0 0 3 】

画面プログラムの単体テストにおいては、従来はテスト仕様書を作成してテス  
トを行うことはあまりなく、画面プログラムの作成者、または単体テストの担当  
者が適当なパターンを作成してプログラムのテストを行うことが多く、品質保証  
の面で問題が残ってしまう。また、テスト仕様書を作成してテストを行う場合に  
も、画面の定義を人間が目視で参照して仕様書を作成することが多く、例えば最

大値や最小値の勘違い、あるいはテストデータの作成ミスなどが起こる可能性があった。

【 0 0 0 4 】

テストの実行時において仕様書に基づいてデータ入力を行うとしても、人間がデータ入力を行うことによって入力ミスが起こる可能性は常に存在する。また入力されたデータが残らない場合には、テスト実行後の検証をすることができなかった。更に入力されたデータに対する処理が、画面プログラム上で正常な処理を通ったか、エラーなどの異常処理を通ったかなどの判定は頻繁であり、テスト実行時の画面プログラムの動きを把握することは困難であった。

【 0 0 0 5 】

画面プログラムは入力項目が多いものが多く、また1つの入力項目についてもその変化範囲が広いためテストパターンが非常に大きくなり、全てのパターンを実行することは人間の手作業では困難である。また再テストを行う場合には、以前のテストに対する入力データが残っていないため、結果的には同じ作業が繰り返され、非効率的になってしまう。マウスカーソルやキーボードの操作を記録、スクリプト化する市販のテスト支援ツールでは、画面の位置や画面部品などの配置が変わるとツールを使用することができず、画面部品などの配置が変わった場合には再テストには使えないという問題点があった。

【 0 0 0 6 】

画面の起動時間や、画面上のボタンを押した場合のレスポンス時間の測定など、プログラムの実行性能を測定する場合には担当者がストップウォッチなどを用いて時間を計測することが多く、また一般に複数回測定して集計を行う必要があり、担当者の作業負荷が大きくなってしまう。画面プログラムをテスト用に修正して処理時間を表示することも不可能ではないが、測定終了後に修正した箇所を元に戻す必要があり、テスト終了後にプログラムソースの変更を行わなければならない。修正ミスなどによってテスト完了後のプログラムの品質に不安が残ることになる。更にテスト実行後のテスト報告書の作成においては、テスト担当者が手作業で記入していくために、記入ミスなどが発生する可能性があるという問題点もあった。



## 【 0 0 0 7 】

本発明の課題は、上述の問題点に鑑み、GUIを使用する画面プログラムのテスト作業全般を効率的、自動化するためのテスト支援装置、およびテスト支援方法を提供することであり、一般にテストの仕様書作成、テストの実行、テストの報告書作成というテスト作業の3段階における様々な問題点を解決することである。

## 【 0 0 0 8 】

## 【課題を解決するための手段】

図1は本発明の原理構成ブロック図である。同図はグラフィックユーザインタフェース（GUI）を使用する画面プログラムのテストを支援するためのGUI系プログラムのテスト支援装置の原理構成ブロック図である。

## 【 0 0 0 9 】

図1においてテスト支援装置1は、テスト支援クラス生成手段2とテスト実行手段3とを備える。テスト支援クラス生成手段2は例えばテスト支援クラス生成装置であり、テスト対象プログラムの画面定義情報を入力として受け取り、そのテスト対象画面プログラムのクラス（スーパークラス、親のクラス）に対してオブジェクト指向の継承の関係におけるサブクラス（子のクラス）であって、画面プログラムをテストするためのテスト支援クラスを生成するものである。

## 【 0 0 1 0 】

テスト実行手段3は、テスト支援クラス生成手段2によって生成されたテスト支援クラスを使用してテスト対象画面プログラムのテストを実行するものである。

## 【 0 0 1 1 】

本発明の実施の形態においては、テスト支援装置1は更にテスト仕様生成手段を備える。テスト仕様生成手段は、画面定義情報からテスト対象画面プログラムに対するテスト仕様を生成するものである。またテスト支援装置1はテスト報告生成手段を更に備えることもできる。テスト報告生成手段は、テスト仕様生成手段によって生成されたテスト仕様と、テスト実行手段3によるテスト対象画面プログラムのテスト実行結果とを用いて、テスト報告を生成するものである。

## 【 0 0 1 2 】

本発明の実施の形態においては、テスト支援クラスは入力テストデータの入力支援を行う機能を備えることも、テストの手動、または自動実行時のテスト結果を記録する機能を備えることも、画面上でのテスト実行箇所の視覚的表示を行う機能を備えることも、また新たな入力テストデータ、または以前のテスト実行結果に対する入力データを用いたテストの手動、または自動実行の機能を備えることもできる。この時、以前のテスト実行結果に対する入力データを用いたテストの実行結果が以前の実行結果と異なる時に警告の表示を行う機能を備えることもできる。更にテスト支援クラスは、テスト対象画面プログラムの実行性能の測定を支援する機能を備えることもできる。

## 【 0 0 1 3 】

本発明のGUI系プログラムのテスト支援方法においては、グラフィックユーザインタフェースを使用する画面プログラムのテストを支援する方法として、画面定義情報を入力として受け取り、テスト対象画面プログラムのクラスに対してオブジェクト指向の継承の関係におけるサブクラス（子のクラス）であって、画面プログラムをテストするためのテスト支援クラスを生成し、生成されたテスト支援クラスを使用してテスト対象画面プログラムのテストを実行する方法が用いられる。

## 【 0 0 1 4 】

更に本発明の計算機読み出し可能可搬型記憶媒体として、グラフィックユーザインタフェースを使用する画面プログラムのテストを支援するための計算機で用いられる記憶媒体において、テスト対象画面プログラムの画面定義情報を受け取るステップと、そのテスト対象画面プログラムのクラスに対してオブジェクト指向の継承の関係におけるサブクラスであって、画面プログラムをテストするためのテスト支援クラスを生成するステップとを計算機に実行させるためのプログラムを格納した記憶媒体が用いられる。

## 【 0 0 1 5 】

また本発明における計算機読み出し可能可搬型記憶媒体として、グラフィックユーザインタフェースを使用する画面プログラムのテストを実行するための計算

機で使用される記憶媒体において、新たな入力テストデータ、または以前のテスト実行結果に対する入力データを受け取るステップと、受け取った入力データを用いて、テスト対象画面プログラムのクラスに対しオブジェクト指向の継承の関係におけるサブクラスであって、画面プログラムをテストするためのテスト支援クラスを使用して、そのテスト対象画面プログラムのテストを実行するステップとを計算機に実行させるためのプログラムを格納した記憶媒体が用いられる。

【0016】

以上述べたように本発明においては、テスト対象画面プログラムのクラスに対してオブジェクト指向の継承の関係におけるサブクラスであって、画面プログラムをテストするためのテスト支援クラスが生成され、そのテスト支援クラスを使用してテスト対象画面プログラムのテストが実行される。

【0017】

【発明の実施の形態】

本発明の基本的な概念について、図2～図6を用いて説明する。図2、および図3は、グラフィックユーザインタフェース（GUI）を使用する画面プログラムのテストを実行するための、オブジェクト指向プログラミングにおける継承の考え方を利用したテスト支援クラスの利用の説明図である。

【0018】

図2で示すように、本発明においては従来のような汎用的な支援ツールを用いたり、テストドライバを生成してテスト実行作業を支援するのではなく、テスト対象画面プログラム10とオブジェクト指向プログラミングにおける継承の関係にある対象テスト支援クラス11が用いられることによって、テスト対象画面プログラム10のテストが実行される。

【0019】

図3は、画面プログラムの拡張によるテスト支援クラスの構成の説明図である。テスト対象画面プログラム10を拡張して、テスト支援機能を付け加えることによって、テスト支援クラス11が構成される。

【0020】

本発明の実施形態においてはテスト仕様書が作成され、前述のテスト支援クラ

スを利用してテストが実行され、その結果に対するテスト報告書が生成される。

図4はテスト仕様書生成の説明図である。同図において画面定義情報15から、テスト仕様書生成装置16によってテストパターンとその入力データファイル17が生成され、その入力データファイルの編集／取り込みが行われて、テスト仕様書18が生成される。テスト仕様書18としては、テスト仕様書生成装置16の出力をそのまま用いてもよい。

#### 【0021】

テストパターンとしては正常な結果が得られると期待されるパターンと、異常な結果、すなわちエラーなどが検出されると期待されるテストパターンなどが用いられ、テスト仕様書18としては例えばエクセルのフォーマットの仕様書が生成されるが、実際にテストを実行するためには、そのフォーマットを例えばデータ項目毎にカンマで区切られているファイルとしてのCSV形式のファイルに変換して、テストを実行するコンピュータが読み込み可能なファイル形式に直す必要がある。

#### 【0022】

図5はテスト支援クラス生成の概念的な説明図である。テスト支援クラス、すなわち図3で説明したようにテスト支援機能が付け加えられた画面プログラムが画面プログラムの定義を元に生成される。すなわち画面定義情報15から、テスト支援クラス生成装置20によってテスト支援クラス21が生成される。画面定義情報15にはテスト対象の画面プログラム10のクラス名（親のクラス名）が記述されており、このクラスを継承することによって子クラスとしてのテスト支援クラス21が生成される。

#### 【0023】

テストの実行時には、生成されたテスト支援クラス21の実行によってテスト対象の画面プログラム10も起動され、テスト支援機能付きの画面が表示される。画面プログラムに付け加えられるテスト支援機能は、次の6つの機能である。

#### 【0024】

第1の機能は、テストパターンおよび入力テストデータの入力支援機能である。この機能を用いることによってテスト仕様書または、生成された入力データフ

ファイルが読み込まれ、画面上の入力項目に対応して入力データの一覧が表示され、テスト作業者がそれらの入力データのいずれかを選択して入力できるように支援を行う。

【 0 0 2 5 】

第 2 の機能は実行結果の記録機能である。テスト実行の日時、テストケースの番号、入力データ、実行結果などをファイルに記録できるようにする機能である。

【 0 0 2 6 】

第 3 の機能はテストの自動実行の機能である。テスト仕様書に基づいた入力データファイル、または以前のテスト実行結果に対する入力データを用いて、入力データを順次自動的にテスト対象の画面プログラムに入力し、その実行結果を記録できるようにする機能である。

【 0 0 2 7 】

第 4 の機能はテスト再実行時の以前の結果との比較機能である。上述の第 3 の機能の自動実行機能において、以前のテスト実行結果に対する入力データを用いてテストを自動、あるいは手動実行する場合に、以前の実行結果と異なる結果が得られら場合に、テスト作業者に対して警告表示できるようにする機能である。

【 0 0 2 8 】

第 5 の機能はテスト実行時のプログラム処理内の通過箇所の視覚的表現機能である。この機能では、画面上のプログラム処理内の通過箇所に対応するコントロール（画面上の部品）の色を、正常な処理が行われた場合と、エラーなどの異常処理が行われた場合とで、例えば青と黄色というように変えて表示することによって、テスト作業者に正常な処理が行われたか否かを知らせることができるようにする機能である。

【 0 0 2 9 】

第 6 の機能はプログラム実行時の性能測定の支援機能である。例えば画面の起動時間や、画面内のボタン押し下げ時のレスポンス時間などを測定し、その結果を表示したり、記録する機能である。プログラムが起動されてから画面が表示されるまでの時間、ボタン押し下げ時の対応する処理終了までの時間などを測定す

るために、その処理の開始／終了時に合わせた時間の測定開始／終了処理を追加して、プログラムの性能を測定できるようにする機能であり、測定結果を画面に表示したり、ファイルに保存したりする機能である。

#### 【 0 0 3 0 】

テストが終了するとテスト報告書の作成が行われる。図 6 に示すように、テスト仕様書 1 8 と実行結果情報 2 3 を用いて、テスト報告書生成装置 2 4 によってテスト報告書 2 5 が生成される。

#### 【 0 0 3 1 】

図 7 はテスト支援装置の全体構成図である。同図は基本的には図 4 ～図 6 をまとめたものであるが、それ以外に画面定義情報 1 5 からテスト対象画面プログラムに相当するテスト対象画面クラス 2 8 を生成する画面クラス生成装置 2 7 を備えている。画面定義情報 1 5 からの画面クラス 2 8 の生成に関しては、本発明と直接の関係がないため、その説明を省略する。

#### 【 0 0 3 2 】

図 7 においてテスト支援クラス 2 1 は、前述の 6 つの機能に対応する 5 つの装置、すなわちテストデータ入力装置 3 1、テストデータ設定装置 3 2、テスト自動実行装置 3 3、性能測定支援装置 3 4、および通過処理の視覚的表現装置 3 5 を備えている。これらの装置の動作については後述する。

#### 【 0 0 3 3 】

図 8 はオブジェクト指向プログラミングにおける継承を利用したテスト支援機能の実現方法の説明図である。同図において、テスト対象画面クラス 2 8 は作業担当者がテストしたい画面プログラムのクラスであり、テスト支援クラス 2 1 とは別のファイルに格納されているために、テスト終了後にはテスト支援クラス 2 1 によるテスト支援機能を切り離し、このクラス 2 8 を実行するだけで実際の画面プログラムの実行が可能となる。

#### 【 0 0 3 4 】

テスト支援クラス 2 1 (サブクラス) はテスト対象画面クラス (スーパークラス) と継承の関係があり、テスト対象画面を継承して、テスト支援機能付きに拡張されたものである。実際のテスト対象画面プログラムのテストは、テスト対象

画面クラス 2 8 を継承したテスト支援クラス 2 1 を使って行われる。テスト作業  
者からはテスト対象画面クラス 2 8 が拡張されているように見える。テスト支援  
クラス 2 1 は、テスト対象画面クラス 2 8 の機能に加えて、テスト支援機能とし  
て前述のような入力支援機能、テスト自動実行機能、性能測定支援機能、実行結  
果記録機能などの機能を持つものである。

## 【 0 0 3 5 】

見掛け上テスト作業者はテスト支援クラス 2 1 のテストを行っているように見  
えるが、ボタンを押し下げ時の処理などはテスト対象画面クラス 2 8 の処理が呼  
ばれて行われるために、テスト対象画面クラス 2 8 のテストを行っていることにな  
る。このようにテスト対象画面クラス 2 8 を継承することによって、テスト対  
象画面クラス 2 8 そのものを修正する必要はなく、テスト対象画面プログラムの  
テストを実行することができる。従来においてはテスト実行のためには、テスト  
対象画面プログラムソースそのものを修正する必要があったが、本発明によって  
テスト対象画面クラス 2 8 の修正の必要はなく、画面上のコントロール（部品）  
の変更などがあっても、テストの実行には全く問題が生じない。従来においては  
、テスト対象画面クラスとテスト支援クラスとを設けるとしても、これらのクラ  
スは全くの別物であり、テスト支援クラスがテスト対象画面クラスを一方的に知  
っているという関係でしかなかった。

## 【 0 0 3 6 】

続いて本発明のテスト支援装置の動作について、画面提示情報の具体例を用い  
て図 9 ～図 1 5 によって説明する。

図 9 は画面定義情報からのテスト仕様書生成の説明図である。ここでは従業員  
の登録画面のテストのための仕様書生成について説明する。同図において、左上  
の従業員登録画面定義には従業員番号からキャンセルまでの 6 つの項目に対する  
変数名、型が定義されており、従業員番号から電話番号までの 4 つの項目につい  
てはその桁数、従業員番号と郵便番号については更にその範囲、すなわち最小値  
と最大値が定義されている。このような従業員登録画面定義から、左下の従業員  
登録の画面が生成される。これについては前述のように説明を省略する。またテ  
スト仕様書生成装置 1 6 によって、従業員登録画面テスト仕様書が生成される。

## 【 0 0 3 7 】

テスト仕様としては、項目単位の仕様と、画面単位の仕様とが生成される。項目単位の仕様は、従業員登録画面定義の1つの項目に対する一般に複数のテストケースから構成される。例えば従業員番号の項目に対するテストケースはTEST-1001-01からTEST-1001-06の6つのテストケースからなり、例えば最初のテストケースは最小値についてもテストするものであり、テストデータは、“000000”となっており、登録用の従業員番号として正常なものであり、このデータを用いた従業員登録は正しく実行されるはずのものである。このような正常系のテスト仕様に加えて、例えば3番目のテストケースに対するテストデータ“-1”は異常系のテスト仕様であり、このようなデータを用いた従業員登録は正しく実行されないはずのものである。

## 【 0 0 3 8 】

画面単位のテスト仕様は登録すべき4つの項目、すなわち従業員番号から電話番号までの4つの項目を1組として、1つの画面単位のテストデータとするものであり、例えばTEST-G001のテストソースはテスト仕様の最初の4桁のデータから構成される。なお従業員登録画面定義において省略されている項目がある場合などは、例えばユーザからの指示に基づいてテスト仕様の生成が行われる。

## 【 0 0 3 9 】

図10はテストデータを従業員登録画面のフィールドに対して埋め込むためのテストデータ提供装置の動作説明図である。テストデータ提供装置は、基本的にテストデータ入力装置31と、テストデータ設定装置32から構成される。テストデータ入力装置31はテスト仕様書を読み込む、テストデータ設定装置32が読み取ることができる形式のテストデータ入力情報40を作成し、テストデータ設定装置32はテストデータ入力情報40を読み込み、その入力情報に基づいて画面上にポップアップメニューを表示し、テスト作業によって選択された値を画面上の入力フィールドに対して埋め込み動作を実行する。

## 【 0 0 4 0 】

図10においては最初のテストデータとして従業員番号に対する“00000



0”が選択され、画面上の従業員番号の入力フィールドに埋め込まれている。前述のように従業員番号は数字のみが正常値であり、その有効値は000000から999999までであり、数字以外、例えば英字や日本語は異常値である。正常値は当然正しく入力フィールドに対して埋め込まれ、正しいデータが埋め込まれたこと、すなわちテストデータが設定されたことが、テストデータとのそのテスト結果などを記録する実行結果情報23に記録される。テスト作業者はマウス操作のみでテストデータの入力を行うことが可能となり、テスト工程の工数を削減することができる。設定が実行されたテストデータについては、次々と削除を行って、未設定のテストデータを一目で確認可能とすることもできる。

## 【0041】

図11はテストデータ提供装置による画面単位のテストデータ設定動作の説明図である。この画面単位のテストデータ設定動作においては、ポップアップメニューに、一般にそれぞれ複数のテスト項目から構成されるテストケースの名前が複数表示され、ここではTEST-G001のテストケースに対する4つのテスト項目、すなわち従業員番号から電話番号までの4つのテストデータが各入力フィールドに正しく埋め込まれている。この画面単位のテストデータ設定においても、設定が実行されたテストケースに対するデータを次々と削除し、設定が実行されていないテストケースを一目で確認できるようにすることも可能である。

## 【0042】

図12はテスト自動実行動作の説明図である。テスト自動実行は基本的にはテストデータ入力装置31と、テスト自動実行装置33によって行われる。テスト仕様書に基づく入力データは、前述のようにテストデータ入力装置31によってテストデータ入力情報40の形式に変換され、テスト自動実行装置33に与えられることも可能である。この前回の実行結果情報41は、以前に行われた実行結果を格納するものであり、その時の入力データと実行結果の両方を格納している。

## 【0043】

図12では自動テストの内容として従業員登録が実行されており、テストケースTEST-G001のテストデータを用いて従業員登録が正しく行われたこと

が画面上に示されている。すなわちテストケース毎の入力フィールドに対する値の設定と、ボタンの押し下げなどが自動的に行われ、その実行結果が実行結果情報 2 3 として保存される。この動作は後述するように中断ボタンが押されるまで、各テストケースに対して次々と実行される。

## 【 0 0 4 4 】

テスト仕様書が項目単位である場合にもテストの自動実行は同様に行われる。前回の実行結果情報 4 1 を利用する場合には、以前の実行結果と比べて今回の実行結果が異なるテストケースに対して警告、ここでは T E S T - G 0 0 3 の結果が異なることが表示されている。

## 【 0 0 4 5 】

このように従業員登録画面上の登録ボタンが押下されると、入力された従業員番号、郵便番号、電話番号などがチェックされ、例えば従業員番号が最小値から最大値までの範囲外であればエラーメッセージが表示れる。入力データが正常値であれば、例えば従業員登録画面を表示しているパソコンと接続されたサーバ側の登録処理が呼び出され、図示しないデータベースの登録が行われる。

## 【 0 0 4 6 】

図 1 3 はプログラムの通過処理の視覚的表現の動作例である。例えばテスト漏れなどを防止するために、画面上の各コントロール（部品）に対してテストの結果が視覚的に分かるようなテスト支援ロジック機能を埋め込んで、例えばコントロールの背景色を正常な場合と異常な場合とで変えることによって、プログラムの通過処理結果の視覚的表現が実現される。

## 【 0 0 4 7 】

各入力フィールドに対しては、エラー処理を通ると入力フィールドの背景色を黄色、正常系を通れば背景色を青とすることによって、処理が正常に行われたか否かが視覚的に表現される。ボタンについては、一度でもボタンの押し下げ処理が行われると、ボタンの背景色を例えば青に変更することによって、ボタンの押し下げが行われたことが表現される。

## 【 0 0 4 8 】

図 1 4 は性能測定支援機能の動作例である。性能測定支援装置 3 4 は、画面定

義情報 1 5 を用いてテスト実行時の各種の性能を測定し、その結果を実行結果情報 2 3 に格納する。

【 0 0 4 9 】

性能測定支援装置 3 4 は、例えばテスト対象画面の起動時には、画面が表示されるまでの時間を起動時間として測定し、その結果を表示する。ここでは従業員登録画面の表示までの時間が 5 6 0 ミリ秒であることが表示されている。

【 0 0 5 0 】

また性能測定支援装置 3 4 は、テスト対象画面上でのボタンの押し下げ動作によって、対応する処理が終わるまでの時間を測定する。処理の終了は指定された項目の文字列変化、または測定終了メソッドの呼び出しによって検出され、それまでの時間の表示と、その時間データの実行結果情報 2 3 への保存が行われる。

【 0 0 5 1 】

性能測定支援装置 3 4 は、テスト対象画面の画面定義情報 1 5 から、例えば従業員登録画面定義の各項目の一覧を取得し、ボタン押し下げ時の処理の終了を検出すべき項目の候補として保持し、それをテスト作業者に測定項目として指定させる。例えば登録ボタンの押し下げによって登録が終了すると、状況のフィールドが空白から「登録しました」に変化するが、性能測定支援装置 3 4 はこの文字列の変化を検出して、登録ボタン押し下げ時の処理が終了したことを検出する。この処理終了までの時間測定は、前述のように測定終了メソッドを呼び出すことによって行うこともできる。

【 0 0 5 2 】

図 1 5 はテスト報告書生成装置の動作例の説明図である。テスト報告書生成装置 2 4 は、前述のようにテストの実行によって得られた実行結果情報 2 3 とテスト仕様書 1 8 を付き合わせ、テスト仕様書 1 8 において未記入であった実行結果および実施日を記入して、テスト報告書 2 5 を生成する。

【 0 0 5 3 】

図 1 6 は以上説明したテスト支援装置の全体動作例の説明図である。同図においては、テストデータ入力情報に対してテストの実行結果、実行日時などを含む実行結果情報と、起動時間、レスポンス時間などを含む性能測定情報とが保存さ

れる。実行結果情報の 1 行目は起動時間をミリ秒単位で示したものであり、2 行目以降はテストデータ入力情報に加えてテストデータを入力した結果表示された値、実行結果が正常であるかエラーであるかを示す○、または×の判定結果、実行日時、データ入力から結果表示までのレスポンス時間（ミリ秒）がアンダーライン付きで示されている。

## 【 0 0 5 4 】

なお、例えば実行結果情報の 4 行目（TEST-I 0 0 0 1 - 0 3 のテストケースの結果）はテスト内容が“最小値 - 1”、テストデータが“- 1”に対する結果であり、アンダーライン部の最初の“0”は入力データが正常なものでなかったことを示している。

## 【 0 0 5 5 】

続いて本発明におけるテスト支援機能の実現方法や、テスト支援装置の処理フローチャートなどについて更に詳細に説明する。図 1 7、図 1 8 はテスト支援機能を実現する場合の従来方式と本発明の方式との説明図である。ここでは前述の従業員登録画面をテスト対象画面とし、登録ボタンの押し下げの時点から登録完了までの時間を測定する性能測定機能追加について考える。

## 【 0 0 5 6 】

図 1 7 は従来方式における実現方法であり、従業員登録画面クラス内の登録ボタン押し下げ時の処理の中に、測定開始メソッドと測定終了メソッドが呼び出されるようにプログラムの追加が行われる。これは図 1 8 の従業員登録画面クラスのソースプログラムと比較すれば明らかである。

## 【 0 0 5 7 】

図 1 7 ではこのように従業員登録画面クラスを修正する必要があり、テスト支援 A クラスと従業員登録画面クラスとは別ものであり、画面も別になっており、テスト支援 A クラスと従業員登録画面クラスとの間には、単に知っているという関係があるにすぎない。なお測定開始メソッドは例えばストップウォッチを起動するメソッドであり、測定終了メソッドはストップウォッチを止めるメソッドである。

## 【 0 0 5 8 】

図 1 8 の本発明の方式においては、従業員登録画面クラスの修正は必要がなく、テスト支援 B クラス内に登録ボタン押下時の処理を行うためのソースプログラムを生成するだけであり、テスト支援 B クラスを実行するだけで従業員登録画面クラスに対するテストが行われ、画面表示もテスト支援 B クラスによって行われる。

#### 【 0 0 5 9 】

なおテスト支援 B クラスでは、ボタンが押されたときの処理メソッド、すなわち `torokuButton-action()` をスーパークラスで呼び出す前後に測定開始と終了のメソッドが埋め込まれている。

#### 【 0 0 6 0 】

図 1 9 はテスト仕様書生成装置の処理フローチャートである。同図において処理が開始されると、まずステップ S 1 で画面定義情報の読み込みが行われ、ステップ S 2 で全ての行の読み込みが行われていたか否かが判定され、まだ行われていなかった場合にはステップ S 3 で項目単位のテスト仕様の生成と仕様書への書き込みが行われ、ステップ S 1 以降の処理が繰り返される。ステップ S 2 で全ての行の読み込みが行われていたと判定されると、ステップ S 4 で画面単位のテスト仕様の生成が行われ、それがテスト仕様書に書き込まれて処理を終了する。

#### 【 0 0 6 1 】

図 2 0 はテスト支援クラス生成装置の処理フローチャートである。同図において処理が開始されると、まずステップ S 6 でコントロール情報、すなわち画面上の部品などに関する情報が取得され、ステップ S 7 で全てのコントロール情報が取得されていたか否かが判定され、まだ取得されていなかった場合にはステップ S 8 でコントロールに対するテスト支援処理機能、例えば前述の登録ボタン押下時の登録処理終了までの時間測定機能の処理などが生成され、ステップ S 6 以降の処理が繰り返される。

#### 【 0 0 6 2 】

ステップ S 7 で全てのコントロール情報が取得されていたと判定されると、ステップ S 9 で画面に対するテスト支援処理機能が生成され、ステップ S 1 0 でテスト支援クラスソースに生成されたテスト支援処理機能が書き出されて処理を終

了する。ここでステップ S 9 ではテストデータ入力装置、テストデータ設定装置、テスト自動実行装置、性能測定支援装置などがテスト支援処理機能として生成され、テスト支援クラス内に設けられる。

#### 【 0 0 6 3 】

テスト支援クラスは基本的には前述のように画面定義情報、すなわちテスト対象画面クラスの画面定義情報から生成される。すなわち画面上のコントロールなどに対応する項目の属性の最大値や、最小値などの情報が画面定義情報から得られて、テスト支援クラスが生成される。

#### 【 0 0 6 4 】

これに対して図 7 で説明したように画面クラス生成装置 2 7 によって生成されたテスト対象画面クラス 2 8 からテスト支援クラスを生成することも可能である。すなわち画面上のコントロールなどについてはテスト対象画面クラス 2 8 から必要な情報を取得し、各項目の属性情報、すなわち最大値や最小値などの情報は、例えば利用者からの指定を受けてテスト支援クラスを生成することもできる。

#### 【 0 0 6 5 】

図 2 1 はテストデータ入力装置の動作例、図 2 2 はその処理フローチャートである。図 2 1 においてテストデータ入力装置 3 1 は、例えば従業員登録画面テスト仕様書（従業員番号） 1 8 を読み込み、例えばエクセル形式の仕様書データをテスト自動実行装置が読み取れる形式のテストデータ入力情報 4 0 に変換して出力する。

#### 【 0 0 6 6 】

テストデータ入力情報 4 0 は画面名、入力フィールド、第一階層表示データ、第二階層表示データ、およびテスト値の 5 つの項目からなっており第一階層表示データはテスト仕様書のテスト項目に、また第二階層表示データはテスト内容に対応する。

#### 【 0 0 6 7 】

図 2 2 のテストデータ入力装置の処理フローチャートにおいて処理が開始されると、まずステップ S 1 2 でテスト仕様書の読み込みが行われ、ステップ S 1 3 で全ての行が読み込まれていたか否かが判定され、読み込まれていなかった場合

にはステップ S 1 4 でテストデータ入力情報への変換とその書き込みが行われ、ステップ S 1 2 以降の処理が繰り返され、ステップ S 1 3 で全ての行が読み込まれていたと判定された時間で処理を終了する。

## 【 0 0 6 8 】

図 2 3 はテストデータ設定装置の動作例を示し、図 2 4 はその処理フローチャートを示す。図 2 3 においてテストデータ設定装置 3 2 は、テストデータ入力情報 4 0 を用いポップアップメニューを画面上に表示し、テスト作業者によって選択されたテスト値を入力フィールドに埋め込み、正しくデータ設定が行われたか否かなどの実行結果情報 2 3 を出力する。実行結果情報 2 3 としては図 1 6 で説明した実行結果情報のうち、起動時間以外の実行結果が出力されている。

## 【 0 0 6 9 】

図 2 4 のテストデータ設定装置の処理フローチャートにおいて処理が開始されると、ステップ S 1 6 でテストデータ入力情報が読み込まれ、ステップ S 1 7 で例えばテスト作業者によって指定された画面名がデータに存在するか否かが判定され、存在しない場合には直ちに処理を終了する。存在する場合にはステップ S 1 8 で指定された入力フィールドが存在するか否かが判定され、存在しない場合にはステップ S 1 6 以降の処理が繰り返される。

## 【 0 0 7 0 】

ステップ S 1 8 で指定された入力フィールドが存在する場合には、ステップ S 1 9 でポップアップメニューが作成されて画面上に表示され、ステップ S 2 0 でテスト作業者によってテスト値、またはテストケースが選択されたか否かが判定され、選択されていない場合にはステップ S 1 9 以降の処理が繰り返される。

## 【 0 0 7 1 】

テスト値またはテストケースが選択されたと判定されると、ステップ S 2 1 でテスト値が入力フィールドに埋め込まれ、ステップ S 2 2 でポップアップメニューが画面上から消去され、ステップ S 2 3 で例えばテスト値が正しく設定されたことが実行結果情報として書き出され、ステップ S 2 4 で実行済みのテスト値の削除を行うか否かが判定され、削除しない場合にはステップ S 1 6 以降の処理が繰り返される。

## 【 0 0 7 2 】

ステップ S 2 4 で実行済みテスト値を削除すべき場合には、ステップ S 2 5 でポップアップメニューとして表示すべきデータから実行済みテスト値が削除されて、ステップ S 1 6 以降の処理が繰り返される。このように実行済みテスト値を削除することによって、前述のように未実行のテスト値、またはテストケースを一目で確認することが可能となる。

## 【 0 0 7 3 】

図 2 5 はテスト自動実行装置の動作例であり、図 2 6 はその処理フローチャートである。テスト自動実行装置 3 3 は、図 1 2 で説明したようにテストデータ入力情報 4 0、または前回の実行結果情報 4 1 を用いてテストの自動実行を行い、その結果を実行結果情報 2 3 として出力する。図 2 5 における実行結果情報は、図 1 6 の実行結果情報と比較して起動時間の測定結果を含まない点と、テストの自動実行のためテスト作業員がそれを見て実行結果が正常か異常かを判定するための表示としての○または×が表示されていない点とが異なっている。

## 【 0 0 7 4 】

図 2 6 において処理が開始されると、まずステップ S 3 0 でテストデータ入力情報または前回の実行結果情報が読み込まれ、ステップ S 3 1 で読み込みが最後まで終了していたか否かが判定され、終了していた場合には直ちに処理を終了する。まだ最後まで読み込まれていなかった場合には、ステップ S 3 2 で例えば作業担当者によって指定された画面名がデータに存在するか否かが判定され、ない場合にも直ちに処理を終了する。

## 【 0 0 7 5 】

指定された画面名が存在する場合には、ステップ S 3 3 で指定された入力フィールドが存在するか否かが判定され、存在しない場合にはステップ S 3 0 からの処理が繰り返される。入力フィールドが存在した場合には、ステップ S 3 4 でテスト値が入力フィールドに埋め込まれ、ステップ S 3 5 で実行結果情報、例えばテスト値が正常に埋め込まれたか否かの情報が書き出され、ステップ S 3 6 で中断ボタンが押されたか否かが判定され、まだ押されていない場合にはステップ S 3 0 以降の処理が繰り返される。本実施形態においては、中断ボタンが押される



まではテストの自動実行が行われるものとし、ステップ S 3 6 で中断ボタンが押されたと判定された時点で処理を終了する。

## 【 0 0 7 6 】

図 2 7 は通過処理の視覚的表現装置の動作例であり、図 2 8 はその処理フローチャートである。図 2 7 において、ステップ S 3 8 で例えばテスト作業者によって入力項目への値の設定が行われると、それを受けて通過処理の視覚的表現装置 3 5 はステップ S 3 9 で入力項目の色の変更、すなわち正常な範囲の入力データの設定が行われたか否かを入力フィールドの背景色を変更することによってテスト作業者に対して表示する。図 2 7 の例では正常なデータの入力時には背景色が青、文字が白で表示されるのに対して、異常データの入力時には背景が黄色、文字が黒で表示される。

## 【 0 0 7 7 】

図 2 8 のフローチャートにおいて処理が開始されると、まずステップ S 4 1 で入力項目に対するテストデータが読み込まれ、ステップ S 4 2 で親の画面プログラム、すなわちオブジェクト指向の継承の関係でテスト支援クラスの親（スーパークラス）となっているテスト対象画面プログラムの入力処理にテストデータが渡され、ステップ S 4 3 でエラー、すなわち入力データの異常が検出されたか否かが判定され、エラーが発生した場合にはステップ S 4 4 で入力フィールドの背景色と文字の色が異常処理表現の色に変更された後に、エラーが発生しない場合にはステップ S 4 5 で背景色と文字の色が正常処理表現の色に変更された後に処理を終了する。

## 【 0 0 7 8 】

図 2 9 は性能測定支援装置の動作説明図であり、図 3 0 はその処理フローチャートである。図 2 9 において性能測定支援装置 3 4 は、例えばテスト対象画面クラス 2 8 の起動時には画面が表示されるまでの時間を計測して、その結果を実行結果情報 2 3 として出力すると共に、例えば従業員登録の場合には登録ボタンが押された時点でテスト対象画面クラス 2 8 のボタン処理の呼び出しを行い、ボタン処理の終了時点で、それまでの経過時間を実行結果情報 2 3 として出力することになる。

## 【 0 0 7 9 】

図 3 0 のフローチャートにおいて処理が開始されると、ステップ S 5 0 でテスト対象画面クラスの起動と、画面の表示までの時間の測定が開始され、ステップ S 5 1 で画面が表示されたか否かが判定され、まだ表示されていない場合にはその判定が続けられ、表示されたと判定された場合にはステップ S 5 2 で起動時間測定が終了したと結果の表示が行われ、実行結果情報が書き出される。

## 【 0 0 8 0 】

続いてステップ S 5 3 で監視項目が変更されたか否かが判定され、変更されている場合にはステップ S 5 4 で監視項目の変更が行われた後にステップ S 5 3 以降の処理が繰り返され、監視項目が変更されていない場合にはステップ S 5 5 で対象画面上のボタン、例えば登録ボタンが押されたか否かが判定され、押されていない場合にはステップ S 5 6 で対象画面が終了したか否かの判定が行われ、終了していない場合にはステップ S 5 3 以降の処理が繰り返され、終了したと判定されると処理を終了する。

## 【 0 0 8 1 】

上述の監視項目の変更について図 3 1 を用いて説明する。図 3 1 は文字列変化の検出による性能測定支援機能における監視項目変化の説明図である。

従業員登録画面において、住所補完ボタンが押下されると住所の一部が補完入力され、登録ボタンが押下されると前述のように状況欄に登録成功／失敗に関する情報が表示される。そこで住所補完ボタンを押下したときは監視項目として住所欄、登録ボタンを押下したときは状況欄を監視して文字列変化を検出する。

## 【 0 0 8 2 】

このように測定対象によって監視項目が異なるため、テスト作業者やユーザの意向に応じて監視項目を変更可能とする必要がある。図 3 1 では画面上の入力フィールドでない任意の箇所でマウスを右クリックすることによりテスト支援カスタマイズ用のポップアップメニューが表示される。

## 【 0 0 8 3 】

このポップアップメニューには監視項目の変更の他に、性能測定方法変更、すなわち前述の測定終了メソッド呼出しによる方法への変更や、図 3 0 のステップ

S 6 0、ステップ S 6 2 で説明する測定上限時間の設定のメニューが表示され、テスト支援機能のカスタマイズが可能である。

【 0 0 8 4 】

図 3 0 に戻り、ステップ S 5 5 で対象画面のボタンが押されたと判定されると、ステップ S 5 7 でボタン処理時間、すなわち例えば登録ボタンが押されてから従業員登録が行われ、状況のフィールドに「登録しました」が表示されるまでの時間の測定が開始され、また同時にテスト対象画面クラス 2 8 内のボタン処理が呼び出され、ステップ S 5 8 でボタン処理時間の測定方法が文字列変化検出か、測定用メソッド呼出しかが判定され、文字列変化検出のときにはステップ S 5 9 で指定された項目、例えば前述の状況の項目に文字列の変化などがあったか否かが判定され、ない場合にはステップ S 6 0 で例えばあらかじめ設定されている上限時間を越えたか否かが判定され、越えていない場合にはステップ S 5 9 以降の処理が繰り返される。

【 0 0 8 5 】

ステップ S 5 8 で測定方法が測定用メソッド呼出しのときには、ステップ S 6 1 で測定終了メソッドが呼ばれたか否かが判定され、まだ呼ばれていないときにはステップ S 6 2 で上限時間を越えたか否かが判定され、まだ越えていない場合にはステップ S 6 1 以降の処理が繰り返される。

【 0 0 8 6 】

ステップ S 5 9 で変化があった場合、ステップ S 6 0、ステップ S 6 2 で上限時間を越えた場合、ステップ S 6 1 で測定終了メソッドが呼ばれたと判定された場合には、いずれもステップ S 6 3 で測定が終了したと実行結果情報の書き出しが行われた後に、ステップ S 5 3 以降の処理が繰り返される。

【 0 0 8 7 】

図 3 2 はテスト報告書生成装置の処理フローチャートである。同図において処理が開始されると、まずステップ S 6 5 でテスト仕様書が読み込まれ、ステップ S 6 6 で全ての行の読み込みが行われていたか否かが判定され、読み込まれていたと判定された場合には直ちに処理を終了する。全ての行がまだ読み込まれていなかった場合には、ステップ S 6 7 でその行のテストに対するテスト結果記録が

読み込まれ、ステップ S 6 8 でテスト結果記録の全ての行がすでに読み込まれていたか否かが判定され、読み込まれていた場合にも直ちに処理を終了する。まだ読み込まれていなかった場合には、ステップ S 6 9 でテストケースが一致するかが判定され、一致しない場合にはステップ S 6 5 以降の処理が繰り返される。一致している場合には、ステップ S 7 0 で実行結果と実施日が記入されて、ステップ S 6 5 以降の処理が繰り返される。

## 【 0 0 8 8 】

図 3 3、図 3 4 は本発明のテスト支援装置におけるテスト支援機能の切り離しの説明図である。テスト対象画面プログラムのテストが終了し、そのプログラムを実際に本番環境で使用する前に、テスト支援機能の切り離しが必要となる。図 3 3 はテスト実行環境における表示画面の説明図であり、テスト支援 B クラスを実行することにより、継承の関係にある従業員登録画面クラスに対するテストが実行される。

## 【 0 0 8 9 】

これに対して図 3 4 は本番環境における画面表示例であり、プログラムを実際に使用する本番環境では従業員登録クラスだけが実行される。

従来においては、図 1 7 で説明したように従業員登録画面のソースプログラムに追加・修正が行われるため、テストが終了した後ではその追加・修正を削除し、本来の機能のみにする必要がある。あるいはテストモードと本番モードを設けて、ソースプログラム内の i f 文などで処理のモードを変える必要がある。このようにすると修正ミスや修正漏れの可能性が常に存在し、またモードを変える場合には、その判定のロジックのテストはテストドライバでは行うことができず、この部分について更にテストが必要となる。

## 【 0 0 9 0 】

最後に本発明を実現するためのプログラム、なわちテスト支援のためのプログラムのコンピュータへのローディングについて図 3 5 を用いて説明する。

以上に述べたような本発明の実施形態を実現するためには、テスト支援クラス生成装置 2 0 によってテスト支援クラス 2 1 を生成し、テスト仕様書生成装置 1 6 によって生成されたテスト仕様書とテスト支援クラス 2 1 を用いてテストを実

行し、テスト報告書生成装置 2 4 によってテスト報告書を生成する必要があるが、これらの動作は全て一般的なコンピュータによって実行されることになる。

## 【 0 0 9 1 】

図 3 5 においてコンピュータ 5 1 は本体 5 4 とメモリ 5 5 によって構成されている。メモリ 5 5 はランダムアクセスメモリ (RAM)、ハードディスク、磁気ディスクなどのメモリであり、このようなメモリに本発明を実現するためのプログラムが格納され、そのプログラムが本体 5 4 によって実行されることによって、本発明のテスト支援装置が実現される。

## 【 0 0 9 2 】

本発明の特許請求の範囲の請求項 1 0 ～請求項 1 2 に対応するプログラムや図 1 9、図 2 0、図 2 2、図 2 4、図 2 6、図 2 8、図 3 0 および図 3 2 のフローチャートに示されたプログラムなどはメモリ 5 5 に格納され、そのプログラムが本体 5 4 によって実行されることになる。

## 【 0 0 9 3 】

またこのようなプログラムは、プログラム提供者側からネットワーク 5 3 を介してコンピュータ 5 1 にロードされることも、または市販され、流通してい可搬型記憶媒体 5 2 に格納され、その可搬型記憶媒体がコンピュータ 5 1 にロードされることによって実行されることも可能である。可搬型記憶媒体 5 2 としてはフロッピーディスク、CD-ROM、光ディスク、光磁気ディスク、MO など、様々な形式の記憶媒体を用いることが可能である。

## 【 0 0 9 4 】

以上において本発明の実施形態を詳細に説明したが、ここで本発明の特徴をまとめると、第 1 にテスト仕様書の生成については、画面定義情報からテスト仕様書そのもの、またはデータの形式変換によって入力テストデータとして使えるテスト仕様書が生成されるために、テスト仕様書の生成が自動化され、あるいは手動部分を含むとしても容易となるため、品質保証のためのドキュメントの整備が行いやすくなる。また画面定義情報から仕様書が生成されるために、テストパターンが広い範囲を網羅できることになり、また入力テストデータの画面定義との整合性などにおいても信頼性が向上する。

## 【 0 0 9 5 】

次にテストの実行については、テスト対象画面に対してテスト支援機能が付け加えられることによって、テストデータの入力ミスがなくなり、テスト実行結果の記録が残るため、テストの信頼性が向上する。またテストの自動実行や性能測定支援によって、テスト作業を効率化することができ、テストの実行時には画面プログラム内の通過箇所が画面の入力フィールドなどの色に反映されるために、テスト作業者が画面プログラムの動作状態を把握しやすくなる。

## 【 0 0 9 6 】

マウ斯卡ーソルやキーボードの操作を記録するのではなく、入力データおよびテスト結果を記録するために、例えば画面の位置やコントロールの配置が変わった後の再テストにおいても、テスト実行に影響は生じない。画面プログラム自体には追加が加えられず、テスト完了後のテスト支援機能の切り離しが容易であり、ソースプログラムについて修正を行う必要がないため、テスト完了後のプログラムに対する品質を保証することができる。

## 【 0 0 9 7 】

更にテストの報告書作成については、テストの結果を自動的にテスト報告書に反映するためにテスト作業者の記入ミスなどを防ぐことができ、テスト報告書の信頼性が向上する。

## 【 0 0 9 8 】

## 【発明の効果】

以上詳細に説明したように、本発明によれば入力データの生成とその入力支援、テストの自動実行、およびテスト結果の記録とテスト報告書への反映が可能となり、画面プログラムの単体テストの作業全般、すなわちテスト仕様書作成、テスト実施、およびテスト結果報告の3段階に渡って作業を効率化することができる。またテスト画面の位置やコントロールの配置が変更されても本発明のテスト支援装置は有効に使用することができ、他のOSに移行した後の再テストなどにも使用できる。

## 【 0 0 9 9 】

テスト対象の画面プログラム自体に修正を行う必要がなく、またテスト後のテ

スト支援機能の切り離しが簡単にできるため、テスト終了後の本番環境への移行が容易となり、確実にプログラムの品質を保証することができ、GUI系プログラムの信頼性向上に寄与するところが大きい。

【図面の簡単な説明】

【図 1】

本発明の原理構成ブロック図である。

【図 2】

本発明におけるテスト対象画面プログラムとテスト支援クラスとの関係を説明する図である。

【図 3】

画面プログラムへのテスト支援機能の付加を説明する図である。

【図 4】

テスト仕様書生成の概念的な説明図である。

【図 5】

テスト支援クラス生成の概念的な説明図である。

【図 6】

テスト報告書生成の概念的な説明図である。

【図 7】

本発明のテスト支援装置の全体的な動作の説明図である。

【図 8】

本発明におけるテスト支援機能の実現方法の説明図である。

【図 9】

画面定義情報からのテスト仕様書生成の具体例の説明図である。

【図 10】

項目単位のテストデータ設定の具体例の説明図である。

【図 11】

画面単位のテストデータ設定の具体例の説明図である。

【図 12】

テスト自動実行の具体例の説明図である。

【図 1 3】

通過処理の視覚的表現の具体例の説明図である。

【図 1 4】

性能測定支援装置の動作例の説明図である。

【図 1 5】

テスト報告書生成装置の動作例の説明図である。

【図 1 6】

テスト支援装置の動作例の全体的な説明図である。

【図 1 7】

テスト支援機能実現方法の従来例の説明図である。

【図 1 8】

図 1 7 に対応する本発明におけるテスト支援機能実現方法の説明図である。

【図 1 9】

テスト仕様書生成装置の処理フローチャートである。

【図 2 0】

テスト支援クラス生成装置の処理フローチャートである。

【図 2 1】

テストデータ入力装置の動作例を説明する図である。

【図 2 2】

テストデータ入力装置の処理フローチャートである。

【図 2 3】

テストデータ設定装置の動作例の説明図である。

【図 2 4】

テストデータ設定装置の処理フローチャートである。

【図 2 5】

テスト自動実行装置の動作例の説明図である。

【図 2 6】

テスト自動実行装置の処理フローチャートである。

【図 2 7】



通過処理の視覚的表現装置の動作例の説明図である。

【図 2 8】

通過処理の視覚的表現装置の処理フローチャートである。

【図 2 9】

性能測定支援装置の動作の説明図である。

【図 3 0】

性能測定支援装置の処理フローチャートである。

【図 3 1】

監視項目変化時の性能測定支援装置の動作例の説明図である。

【図 3 2】

テスト報告書生成装置の処理フローチャートである。

【図 3 3】

テスト環境におけるテスト支援 B クラスによるテスト支援の説明図である。

【図 3 4】

プログラム実行の本番環境における従業員登録画面を示す図である。

【図 3 5】

本発明の実施形態を実現するためのプログラムのコンピュータへのローディングを説明する図である。

【符号の説明】

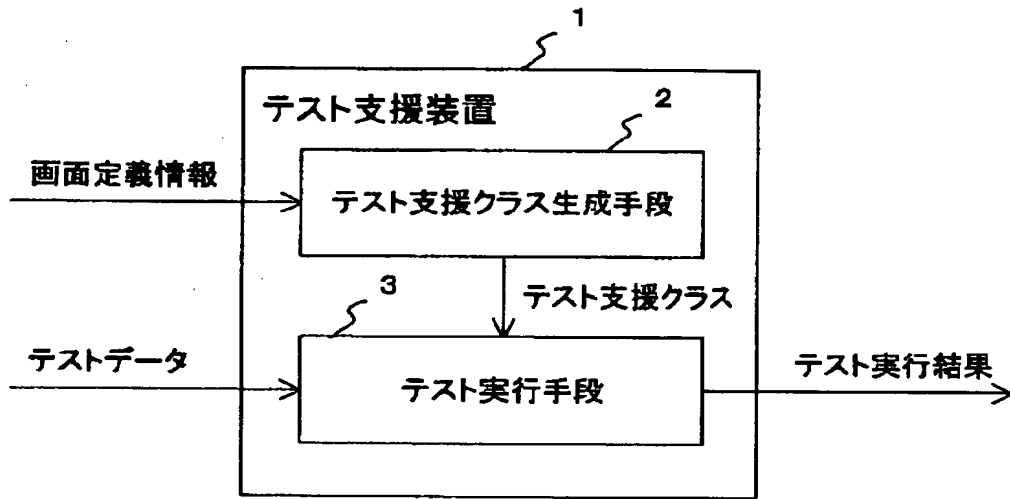
- 1 テスト支援装置
- 2 テスト支援クラス生成手段
- 3 テスト実行手段
- 1 0 画面プログラム
- 1 1, 2 1 テスト支援クラス
- 1 5 画面定義情報
- 1 6 テスト仕様書生成装置
- 1 8 テスト仕様書
- 2 0 テスト支援クラス生成装置
- 2 3 実行結果情報

- 2 4 テスト報告書生成装置
- 2 5 テスト報告書
- 3 1 テストデータ入力装置
- 3 2 テストデータ設定装置
- 3 3 テスト自動実行装置
- 3 4 性能測定支援装置
- 3 5 通過処理の視覚的表現装置
- 4 0 テストデータ入力情報
- 4 1 前回の実行結果情報

【書類名】 図面

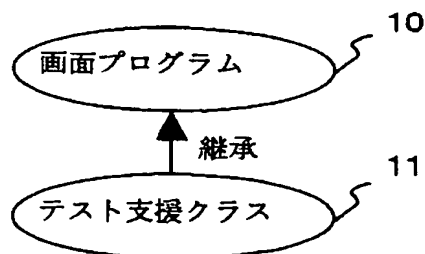
【図 1】

本 発 明 の 原 理 構 成 ブ ロ ッ ク 図



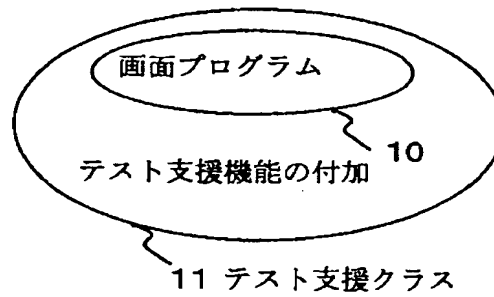
【図 2】

本発明におけるテスト対象画面プログラムと  
テスト支援クラスとの関係を説明する図



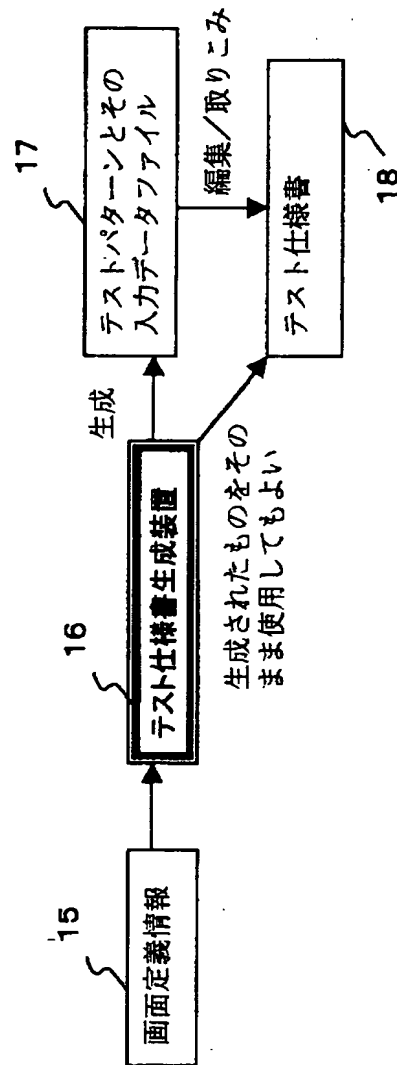
【図 3】

画面プログラムへのテスト支援機能の  
付加を説明する図



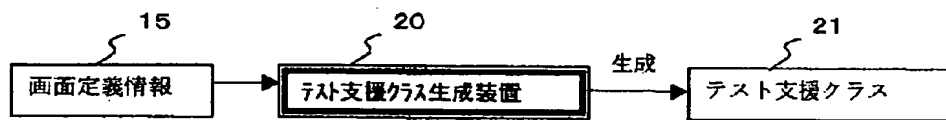
【図 4】

テスト仕様書生成の概念的な説明図



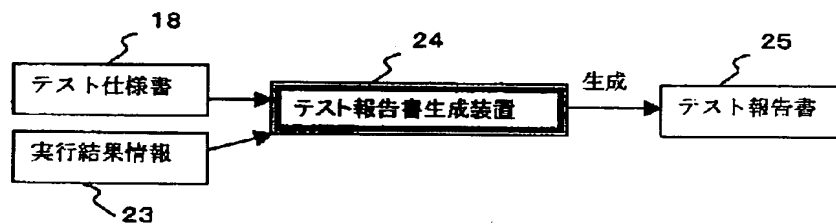
【図 5】

テスト支援クラス生成の概念的な説明図



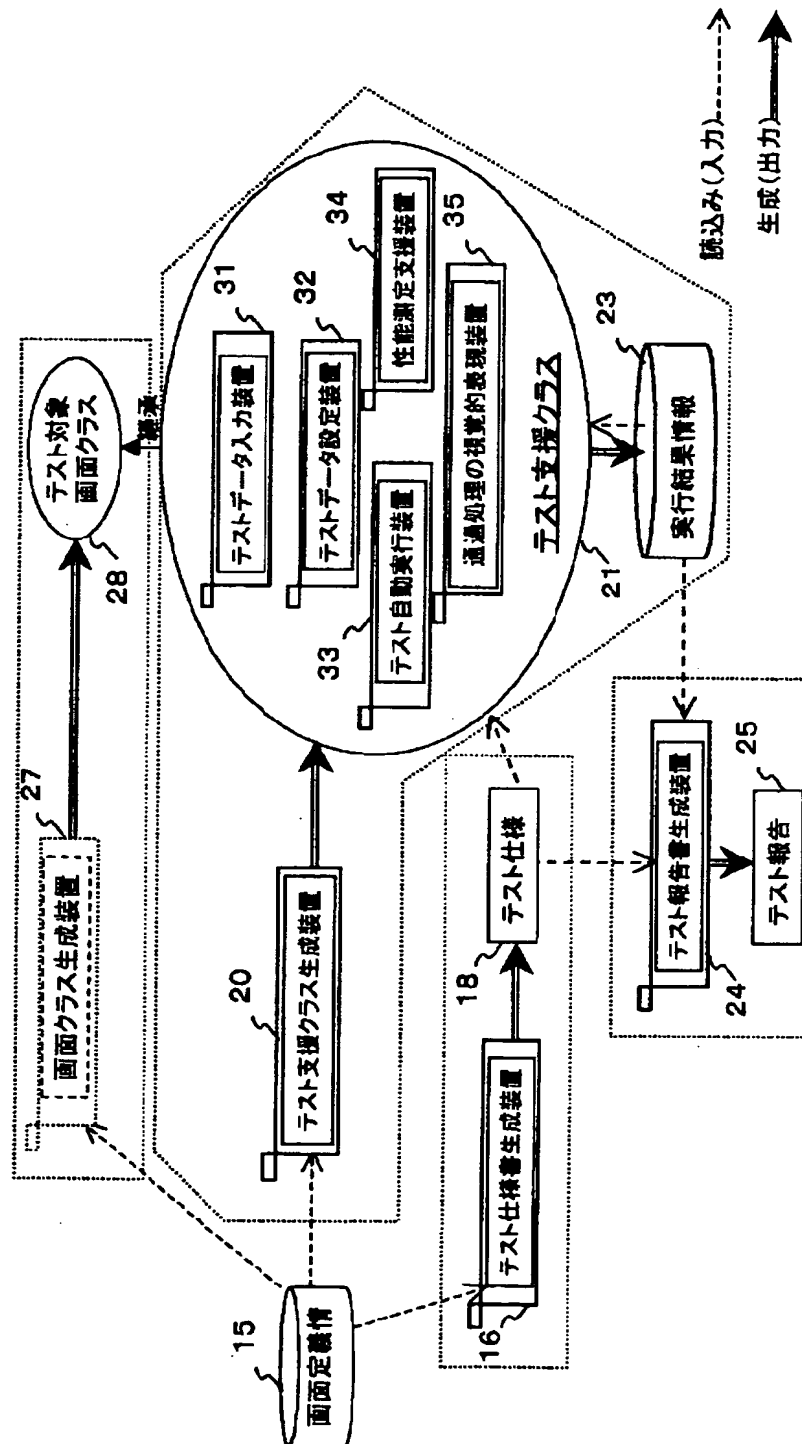
【図 6】

テスト報告書生成の概念的な説明図



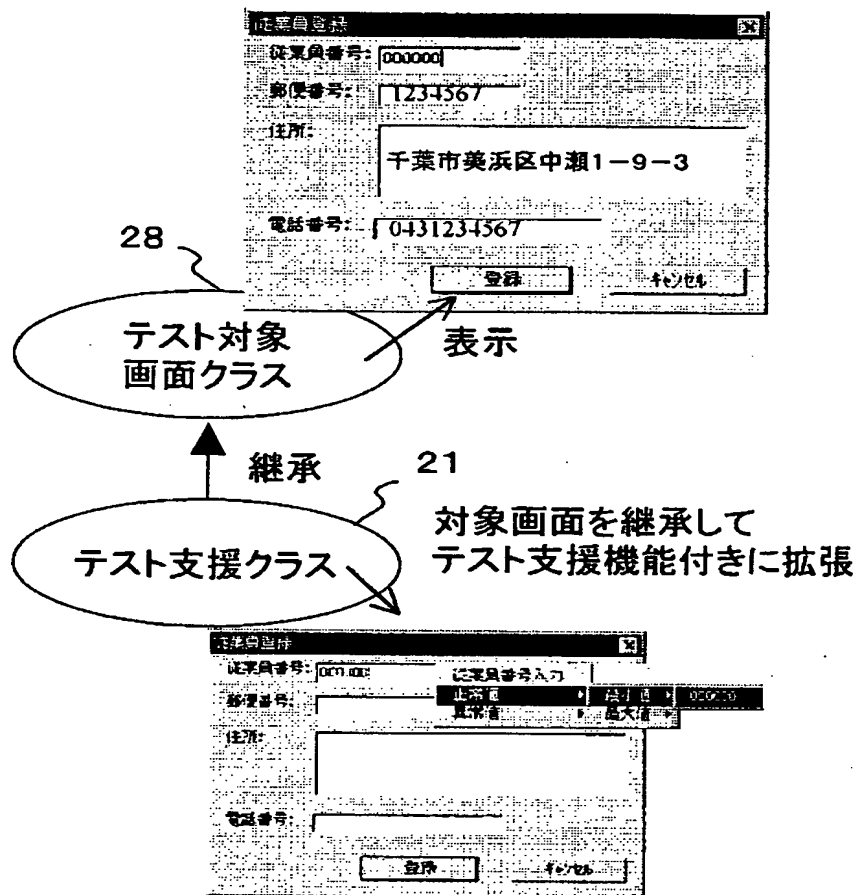
【图7】

## 本発明のテスト支援装置の全体的な動作の説明図



【図 8】

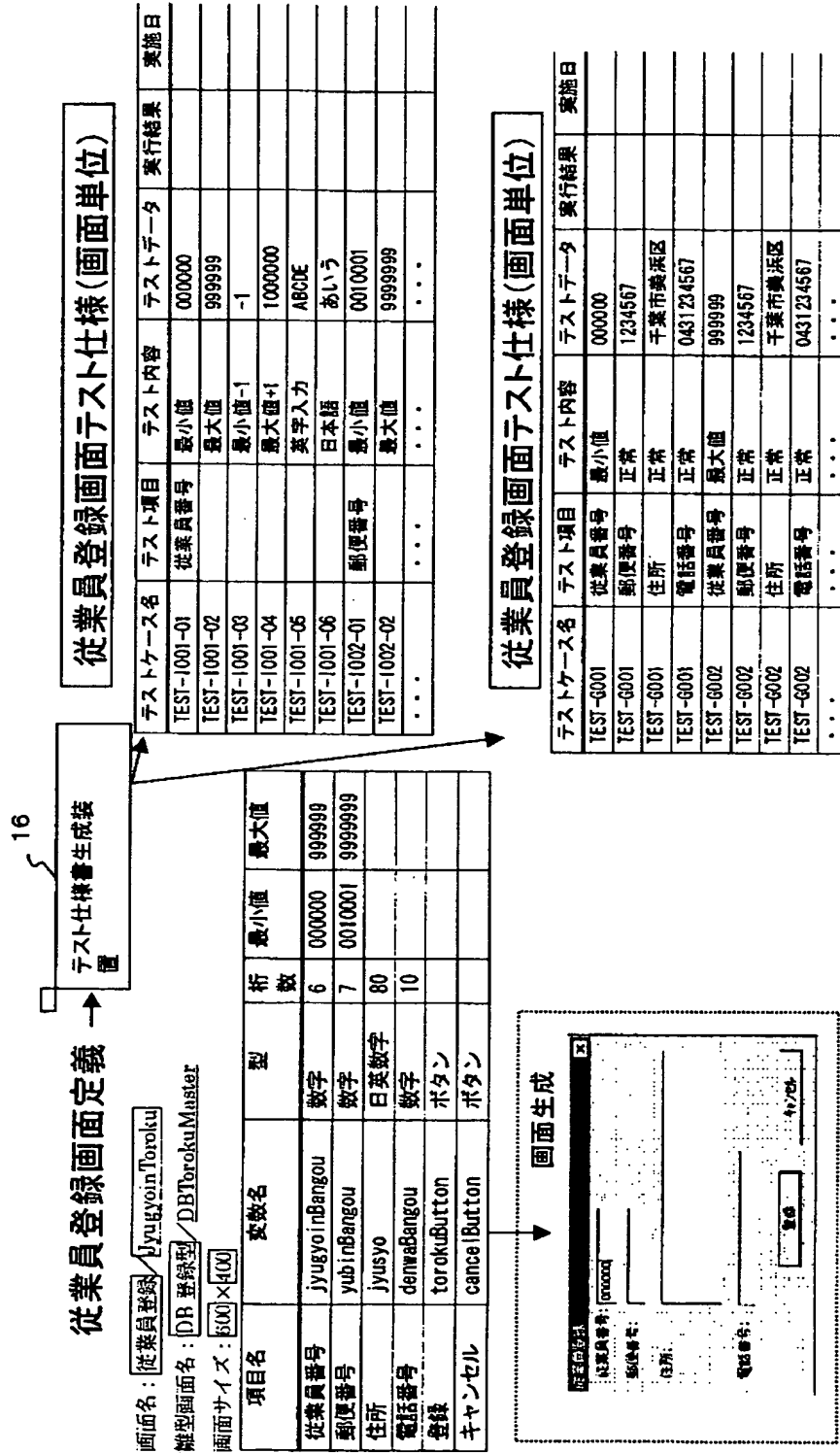
本発明におけるテスト支援機能の実現方法の説明図





【図 9】

画面定義情報からのテスト仕様書生成の具体例の説明図



【図 10】

項目単位のテストデータ設定の具体例の説明図

従業員登録画面テスト仕様書(従業員番号)

テストケース名	テスト項目	テスト内容	テストデータ	実行結果	実施日
TEST-1001-01	従業員番号	最小値	000000		
TEST-1001-02		最大値	999999		
TEST-1001-03		最小値-1	-1		
TEST-1001-04		最大値+1	1000000		
TEST-1001-05		英字入力	ABCDE		
TEST-1001-06		日本語	あいう		
TEST-1002-01	郵便番号				
TEST-1002-02					

従業員登録

従業員番号: [000000]

郵便番号: [ ]

住所: [ ]

電話番号: [ ]

従業員番号入力

正常値

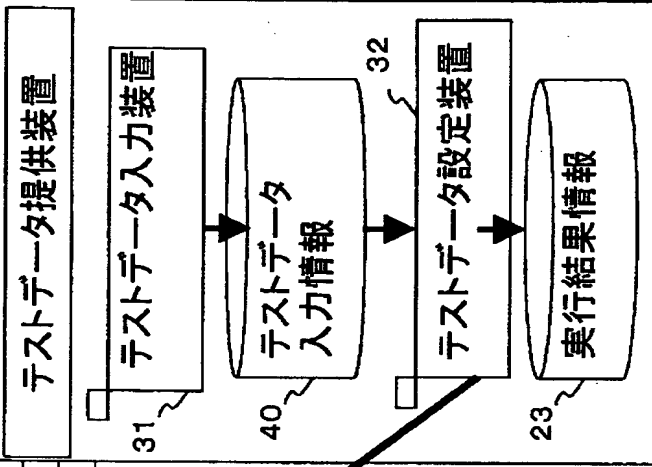
異常値

最小値

最大値

登録

キャンセル

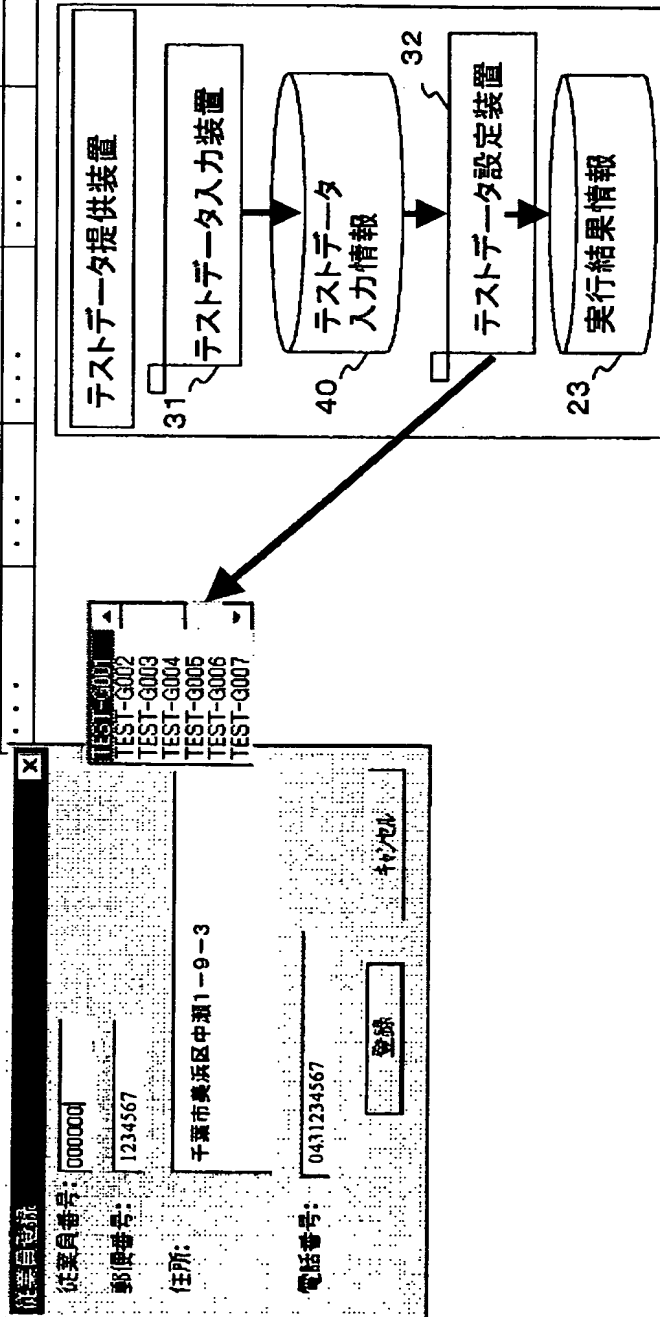


【図 1 1】

画面単位のテストデータ設定の具体例の説明図

従業員登録画面テスト仕様書(従業員番号)

テストケース名	テスト項目	テスト内容	テストデータ	実行結果	実施日
TEST-G001	従業員番号	最小値	000000		
TEST-G001	郵便番号	正常	1234567		
TEST-G001	住所	正常	千葉市美浜区		
TEST-G001	電話番号	正常	0431234567		
TEST-G002	従業員番号	最大値	999999		
TEST-G002	郵便番号	正常	1234567		
TEST-G002	住所	正常	千葉市美浜区		
TEST-G002	電話番号	正常	0431234567		
...	...	...	...		



【図 1 2】

テスト自動実行の具体例の説明図

従業員登録画面テスト仕様書（従業員番号）

テストケース名	テスト項目	テスト内容	テストデータ	実行結果	実施日
TEST-G001	従業員番号	最小値	000000		
TEST-G001	郵便番号	正常	1234567		
TEST-G001	住所	正常	千葉県美浜区		
TEST-G001	電話番号	正常	0431234567		
TEST-G002	従業員番号	最大値	999999		
TEST-G002	郵便番号	正常	1234567		
TEST-G002	住所				
TEST-G002	電話番号				
...	...				

従業員登録

従業員番号: 000000

郵便番号: 1234567

住所: 千葉県美浜区中瀬1-9-3

電話番号: 0431234567

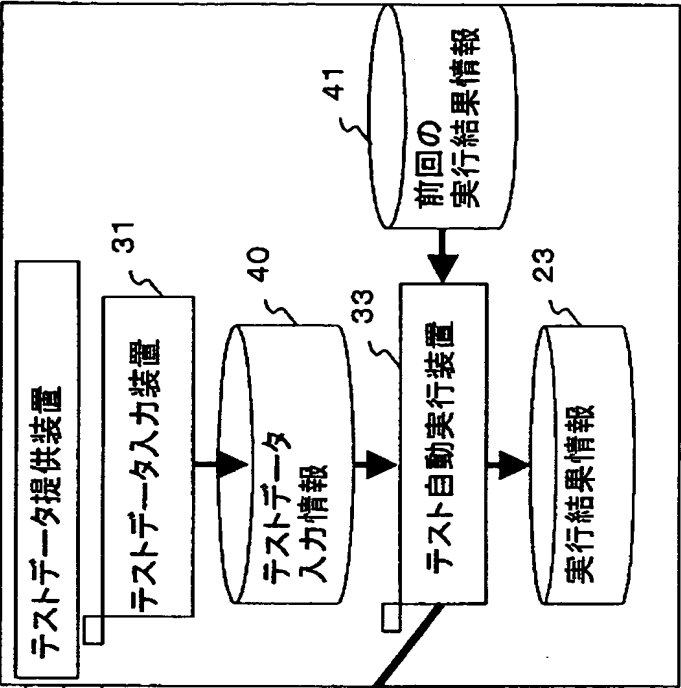
状況: 登録しました

テスト自動実行

TEST-G001  
TEST-G002  
TEST-G003  
TEST-G004  
TEST-G005  
TEST-G006  
TEST-G007

中断

TEST-G003の結果が前回と異なります。登録



【図13】

通過処理の視覚的表現の具体例の説明図

**入カフィールド**

- ・エラー処理を通ると、入カフィールドの背景色を黄色に変更する。
- ・正常系を通ると入カフィールドの背景色を青に変更する。

**ボタン**

- ・一度でも押下するとボタンの背景色を青に変更する。

**従業員登録**

従業員番号: 000000

郵便番号: 1234567

住所: 千葉県美浜区中瀬1-9-3

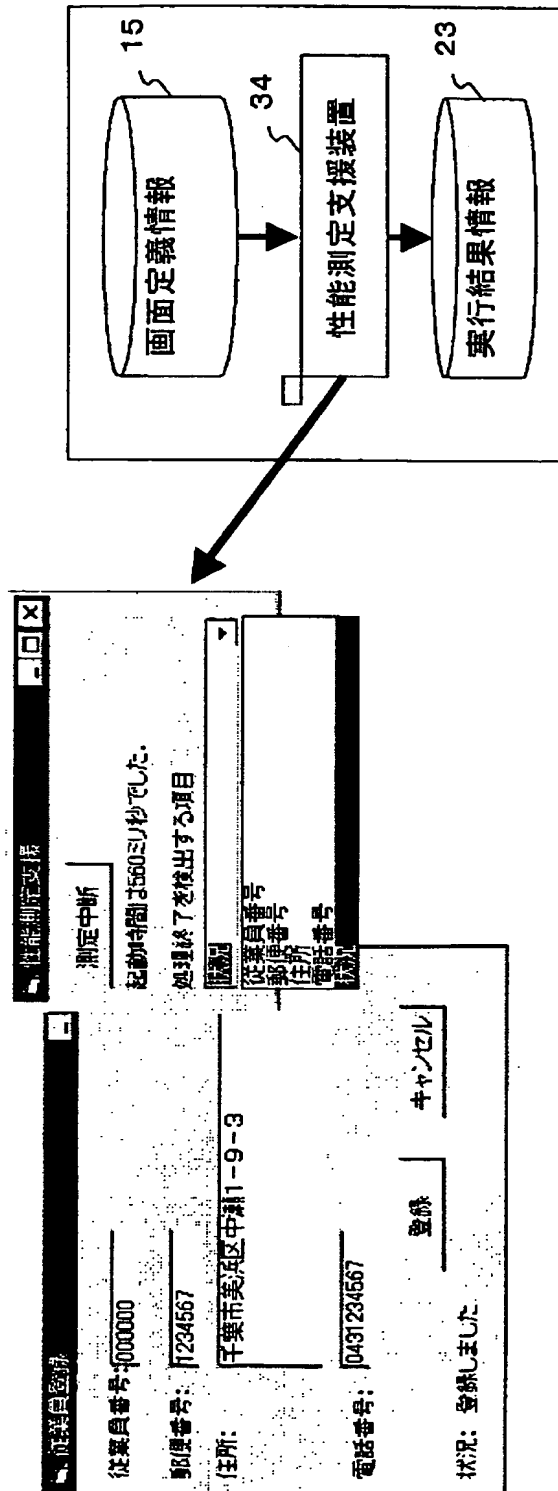
電話番号: 0431234567

登録 キャンセル

状況: 登録しました

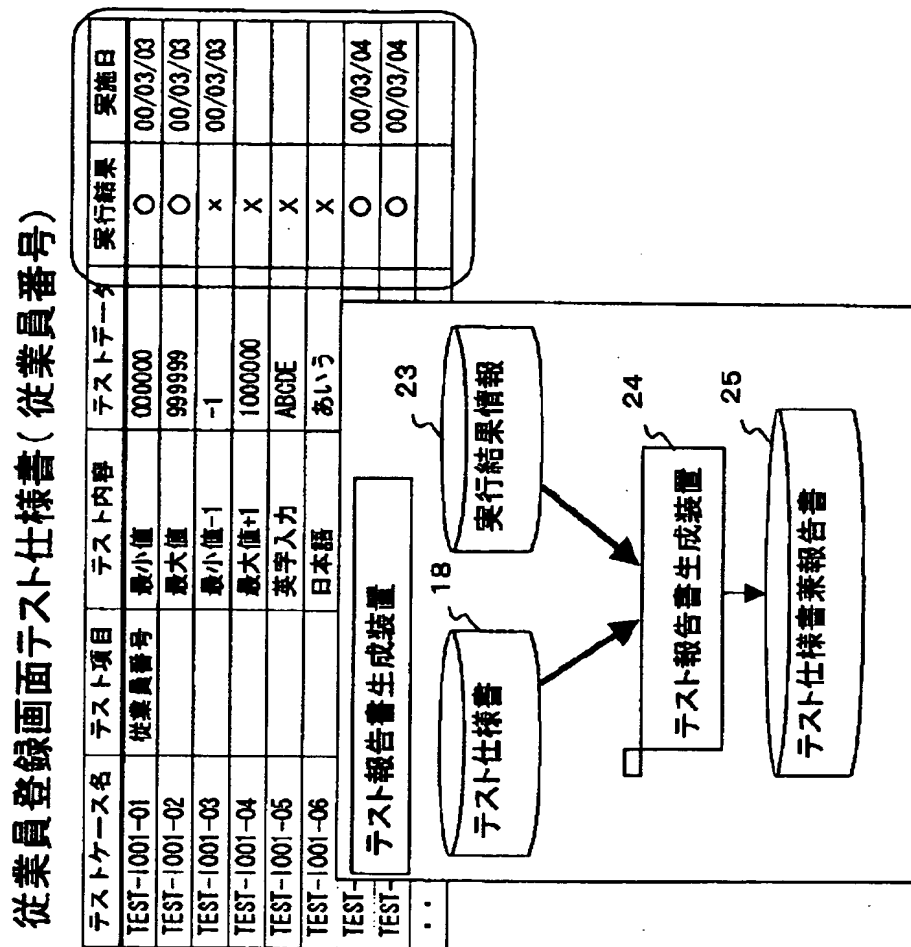
【図14】

性能測定支援装置の動作例の説明図



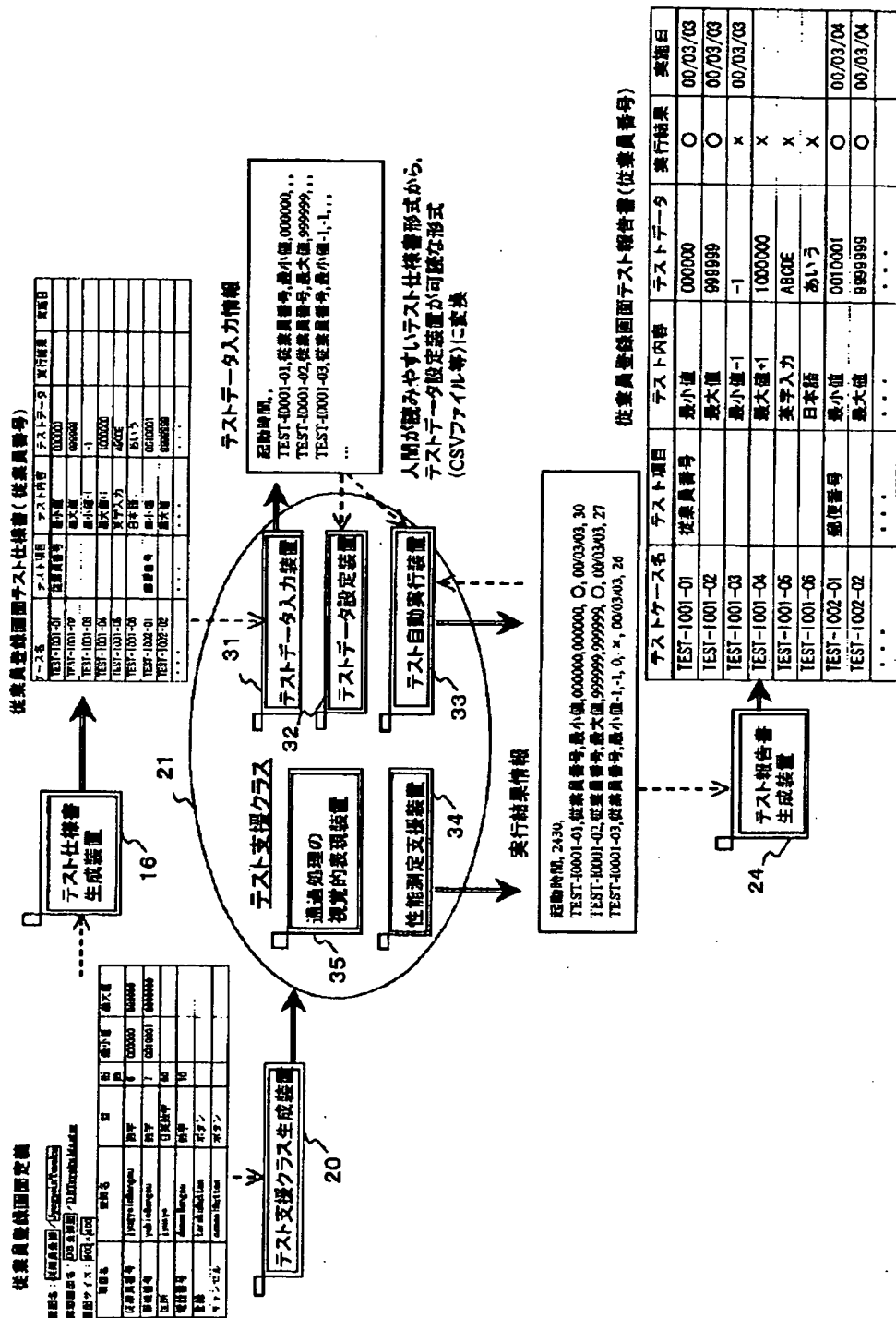
【図 15】

テスト報告書生成装置の動作例の説明図



【图 16】

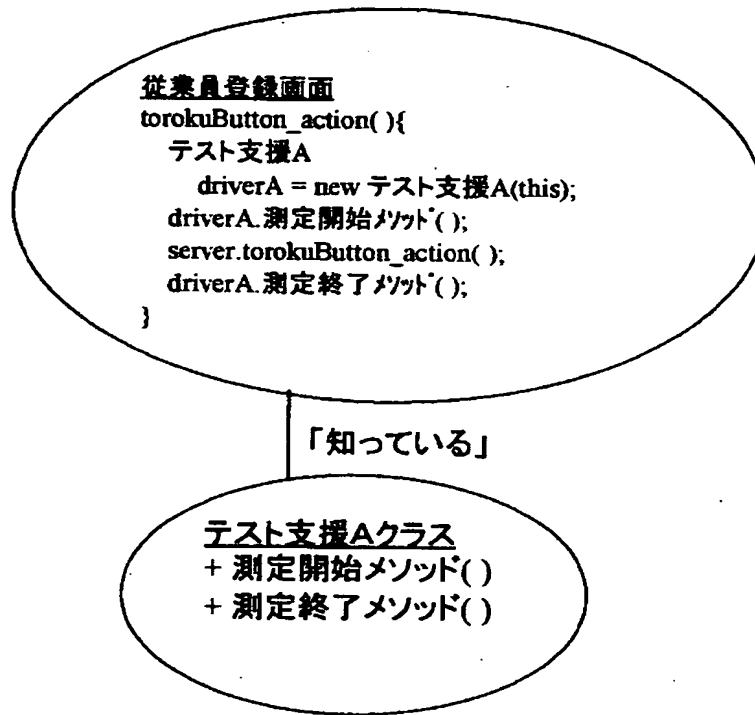
### テスト支援装置の動作例の全体的な説明図





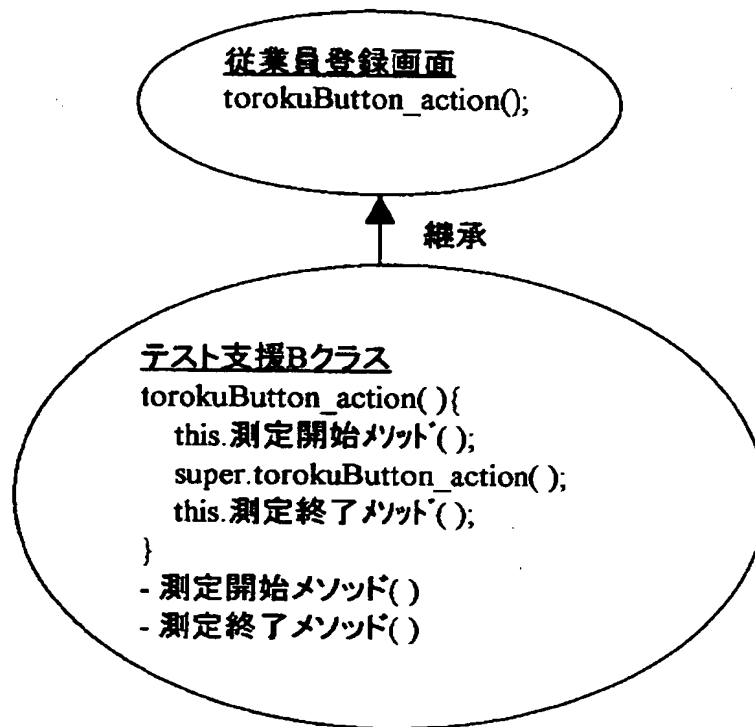
【図 17】

テスト支援機能実現方法の従来例の説明図



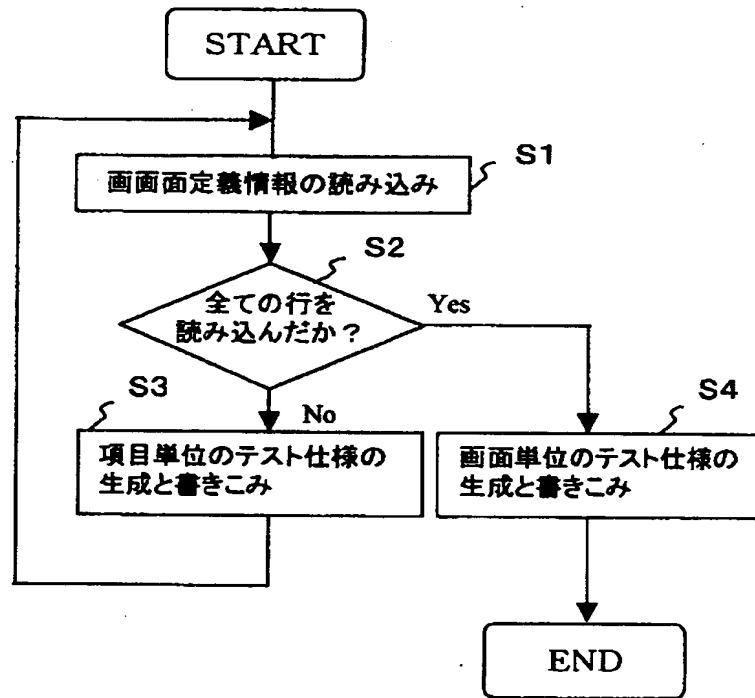
【図 1 8】

図17に対応する本発明における  
テスト支援機能実現方法の説明図



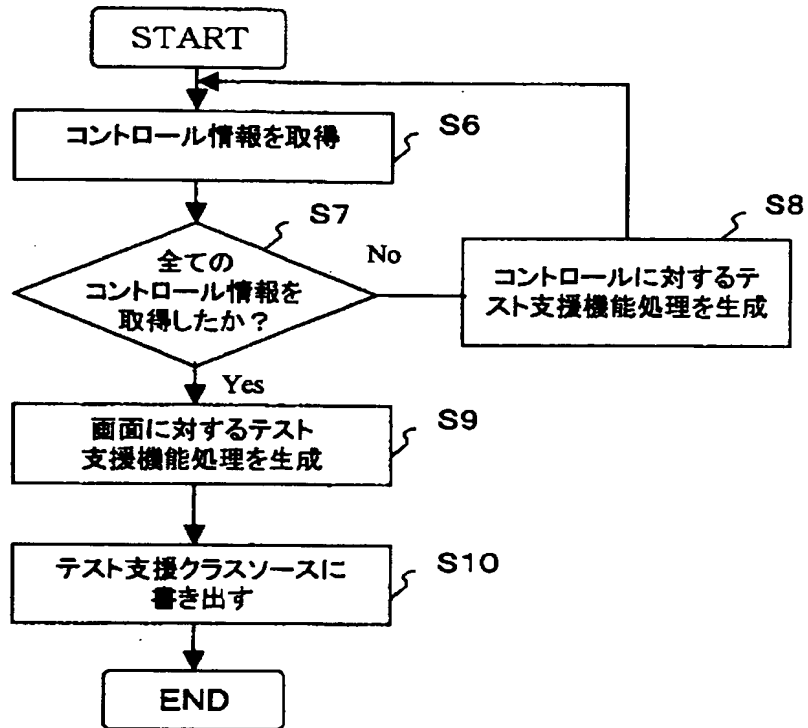
【図19】

テスト仕様書生成装置の処理フローチャート



【図 20】

テスト支援クラス生成装置の処理フローチャート



【図 21】

テストデータ入力装置の動作例を説明する図

従業員登録画面テスト仕様書(従業員番号)<sup>18</sup>

テストケース名	テスト項目	テスト内容	テストデータ	テスト結果
TEST001	正常値	最小値	000000	
		最大値	999999	
	異常値	最小値-1	-1	
		最大値+1	1000000	
		英字入力	ABCDE	
		日本語	あいう	

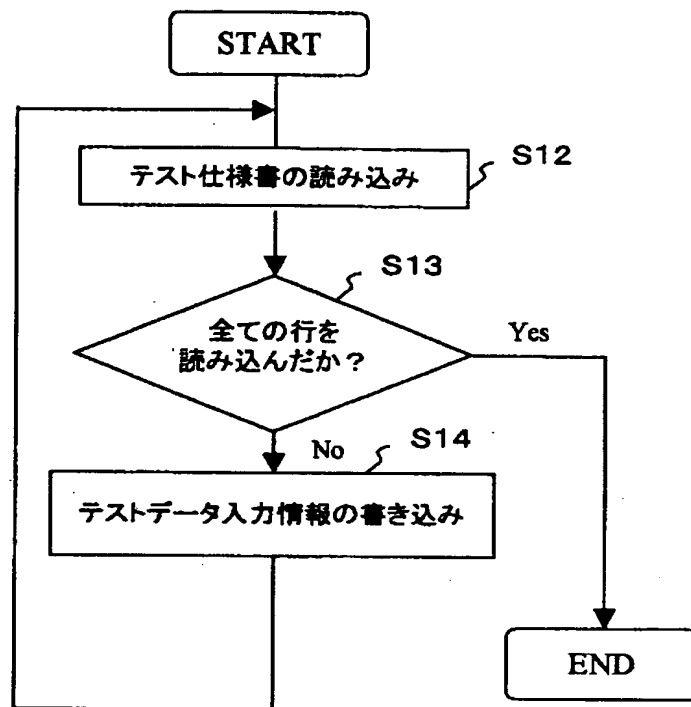
テストデータ入力装置<sup>31</sup>

テストデータ入力情報<sup>40</sup>

画面	入力	第一階層表示	第二階層表示	テスト値
従業員	従業員番	正常値	最小値	000000
従業員	従業員番	正常値	最大値	999999
従業員	従業員番	異常値	最小値-1	-1
従業員	従業員番	異常値	最大値+1	1000000
従業員	従業員番	異常値	英字	ABCDE
従業員	従業員番	異常値	日本語	あいう

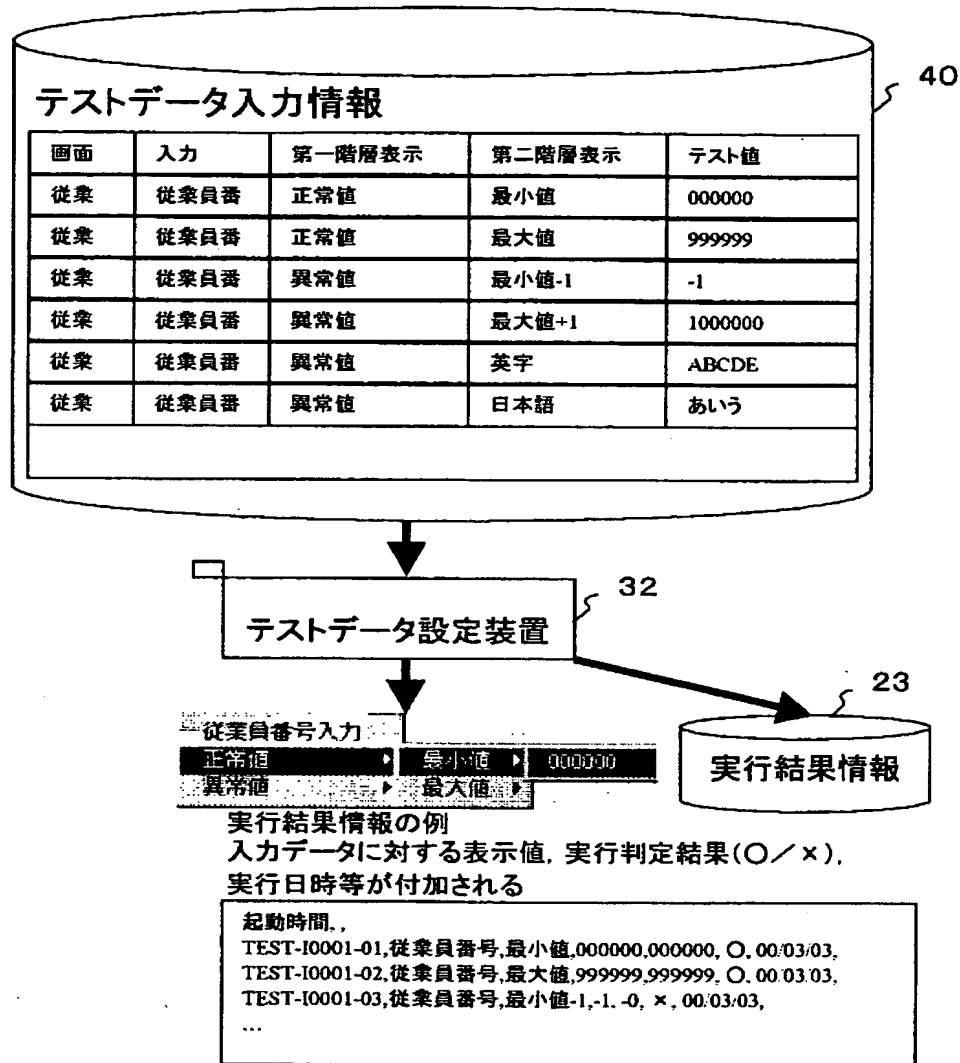
【図 2 2】

テストデータ入力装置の処理フローチャート



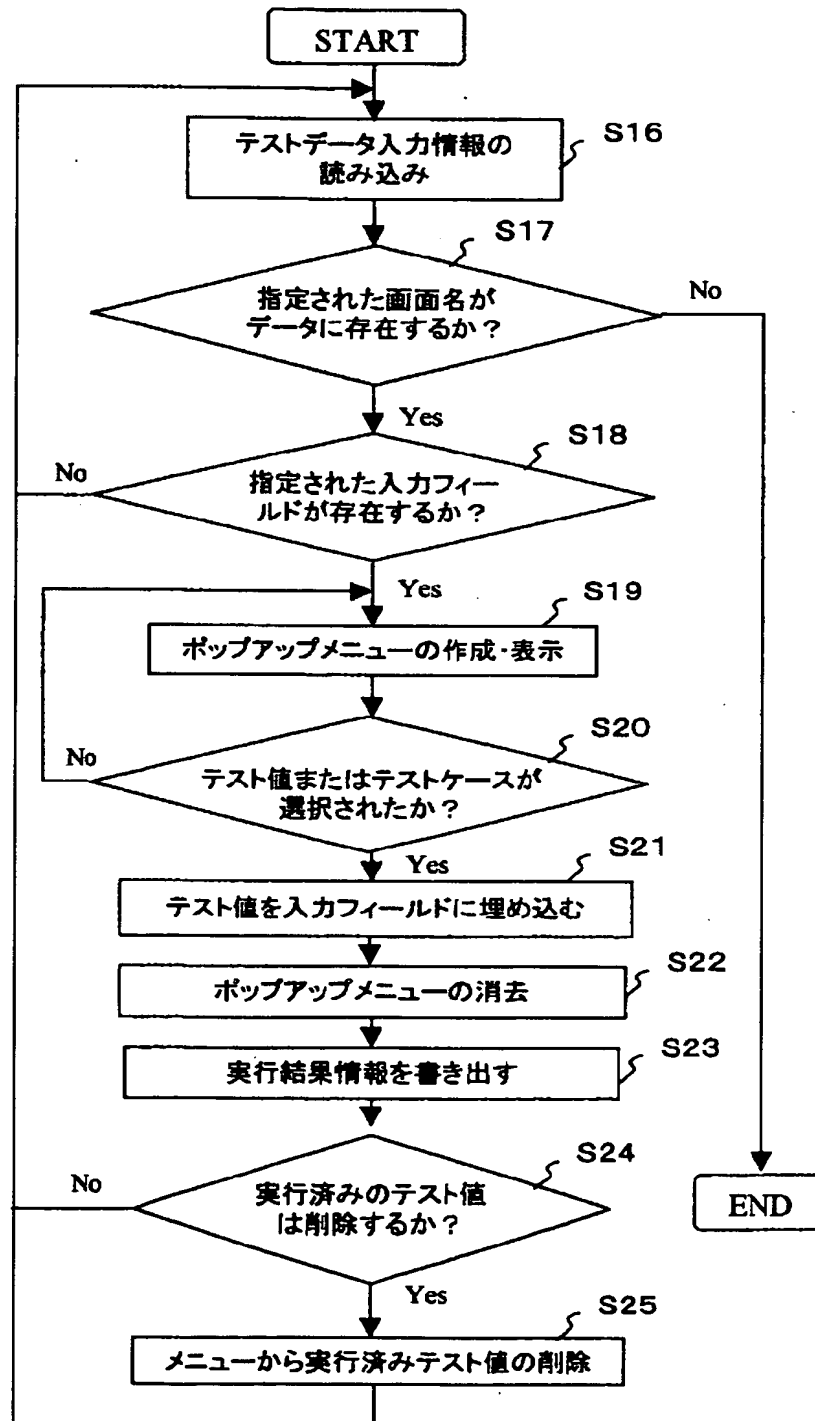
【図 2 3】

テストデータ設定装置の動作例の説明図



【図 24】

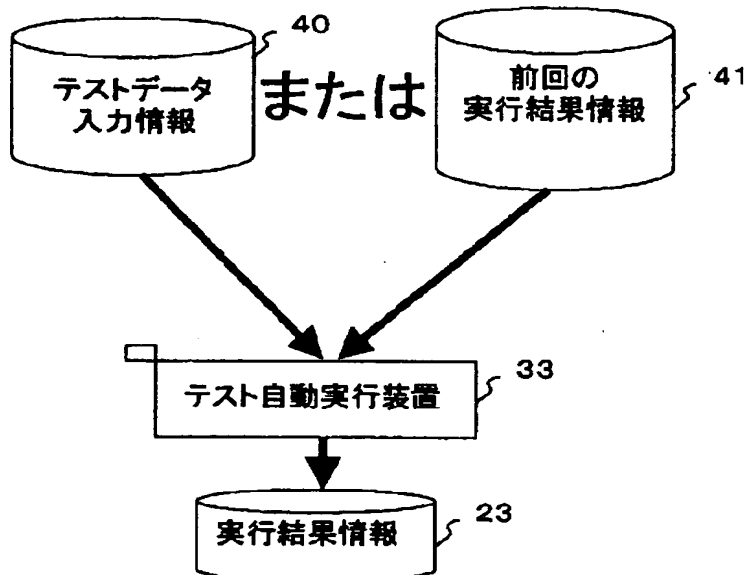
テストデータ設定装置の処理フローチャート





【図 2 5】

テスト自動実行装置の動作例の説明図



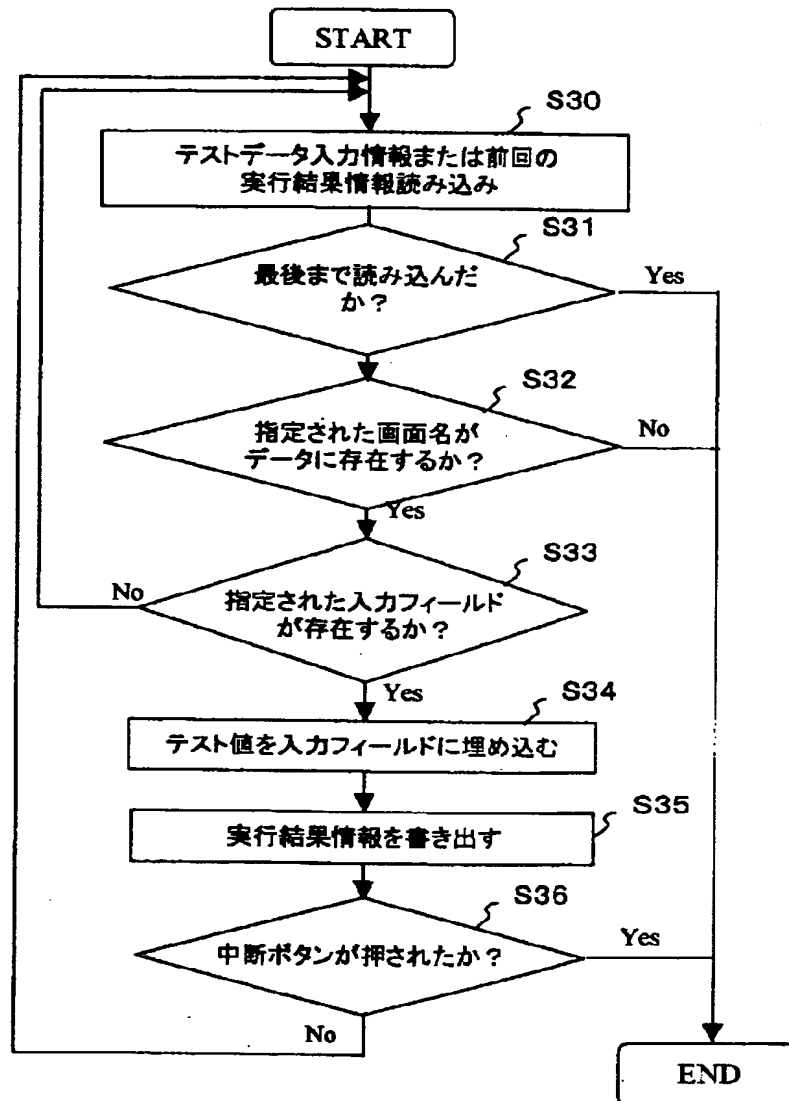
実行結果情報の例

自動実行のため、人間が判断する実行結果判定(○/×)が無い

起動時間,  
 TEST-I0001-01,従業員番号,最小値,000000,000000,,00/03/03,  
 TEST-I0001-02,従業員番号,最大値,999999,999999,,00/03/03,  
 TEST-I0001-03,従業員番号,最小値,-1,-1,0,,00/03/03,  
 ...

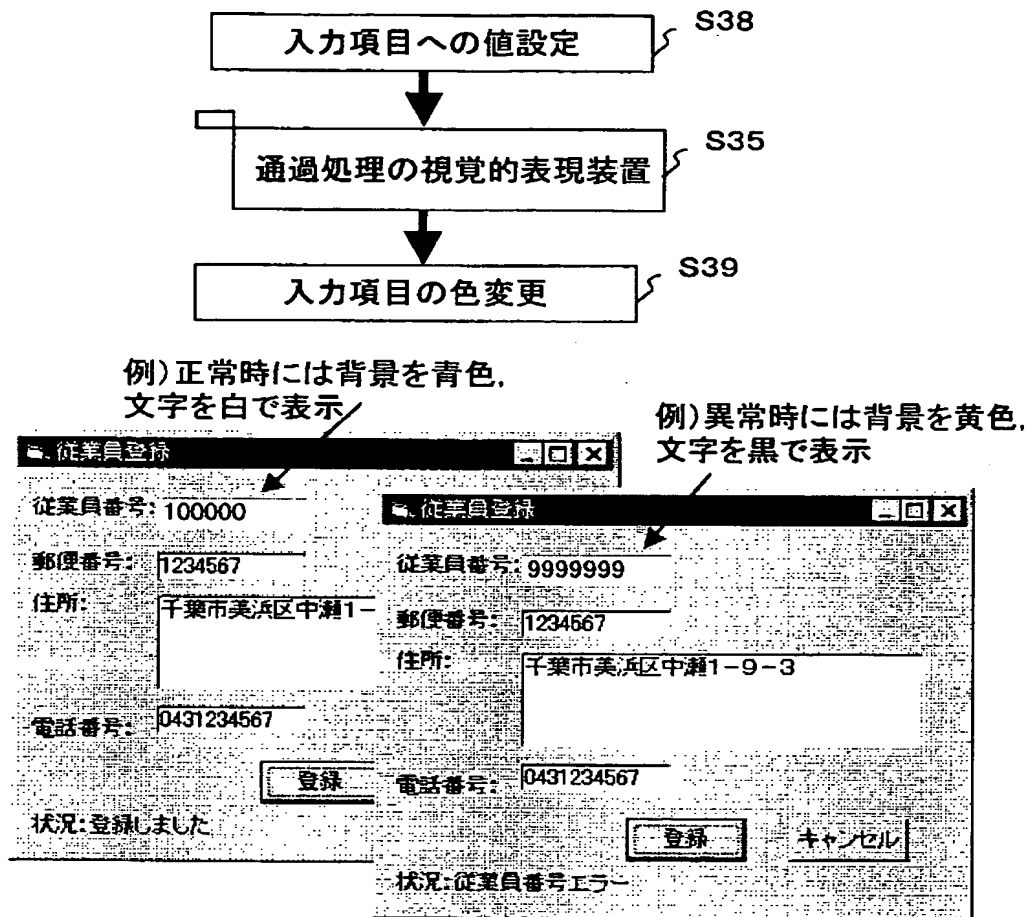
【図 26】

テスト自動実行装置の処理フローチャート



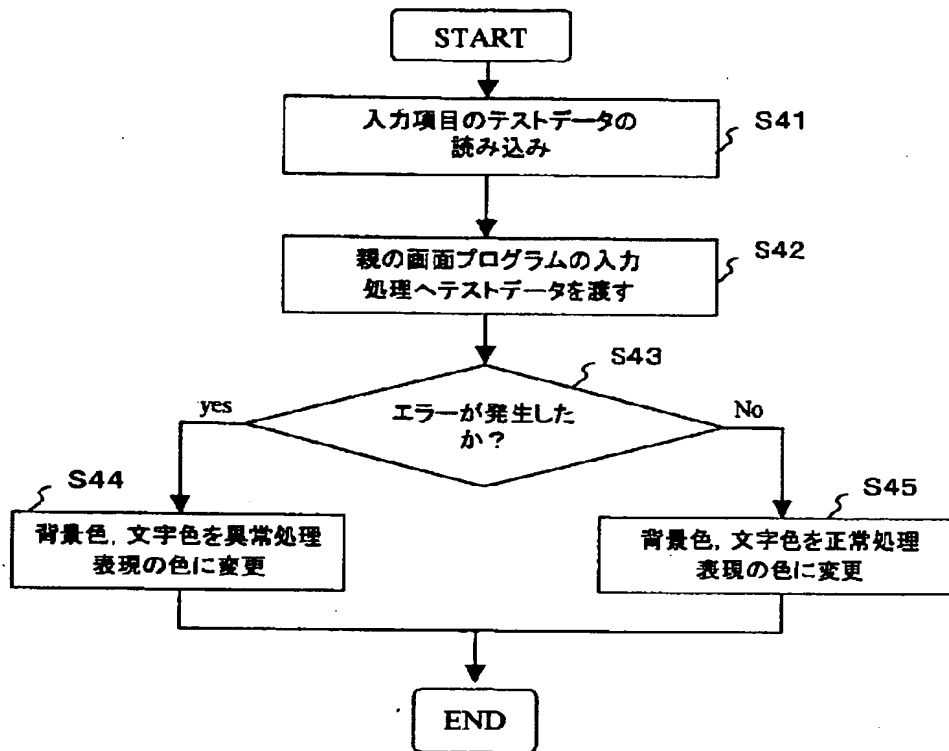
【図 27】

通過処理の視覚的表現装置の動作例の説明図



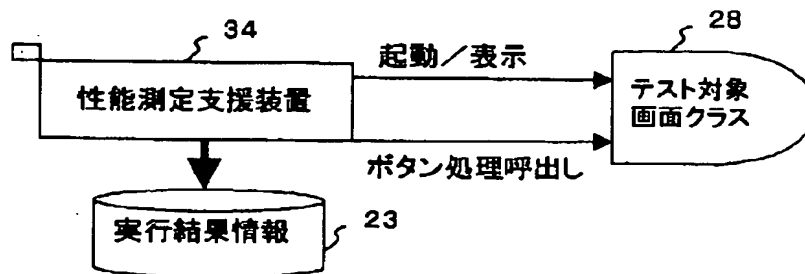
【図 2 8】

通過処理の視覚的表現装置の処理フローチャート



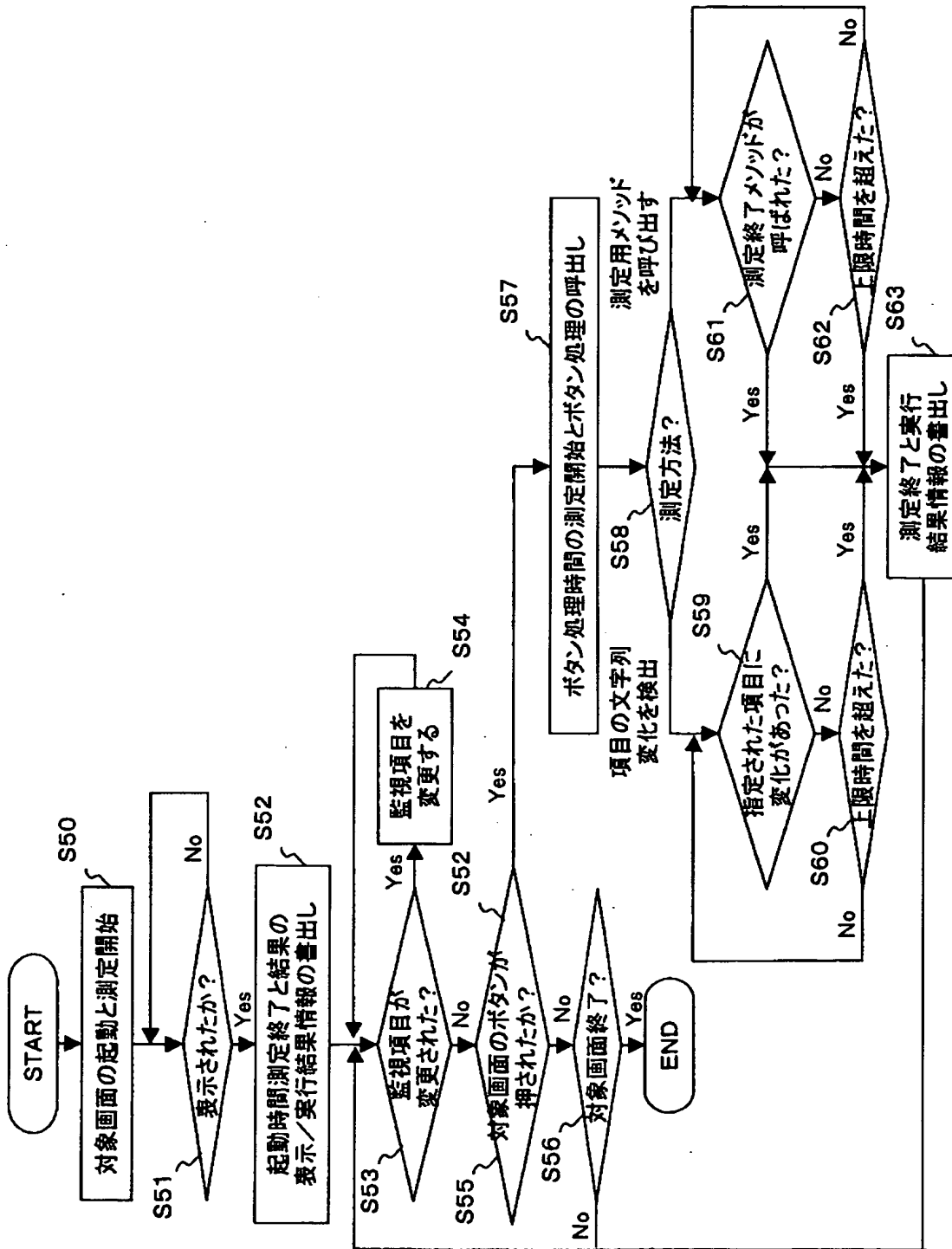
【図 2 9】

性能測定支援装置の動作の説明図



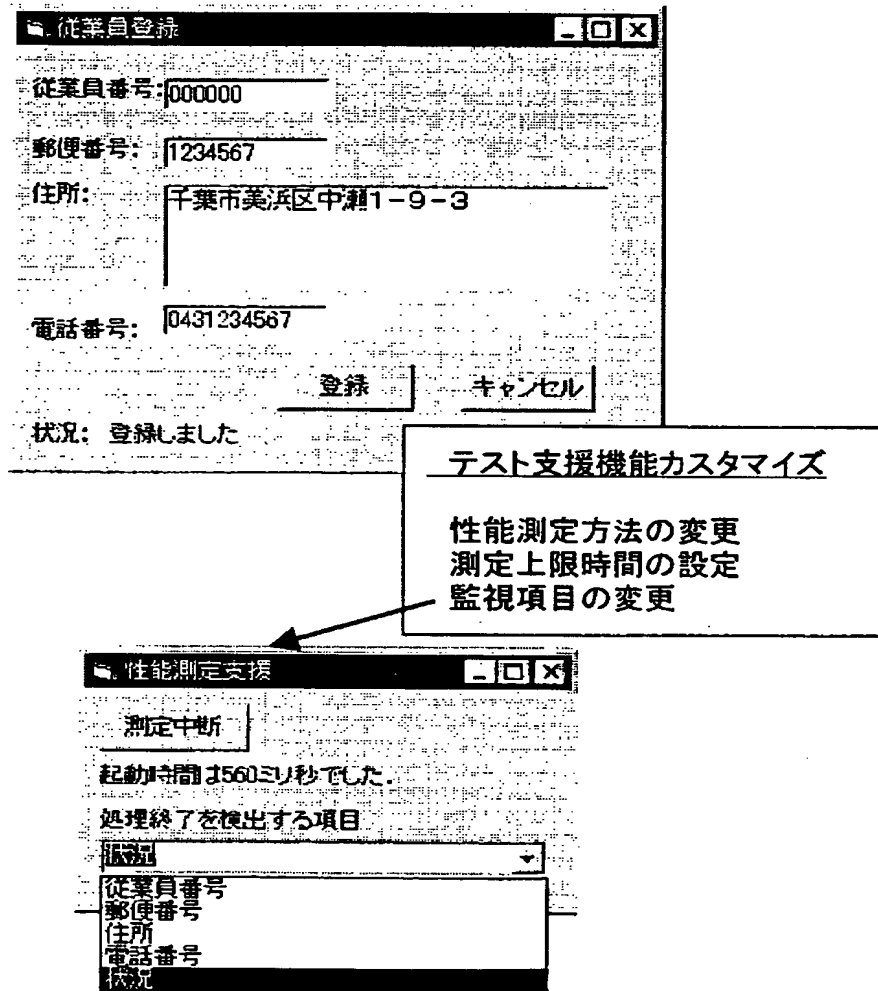
【図 30】

性能測定支援装置の処理フローチャート



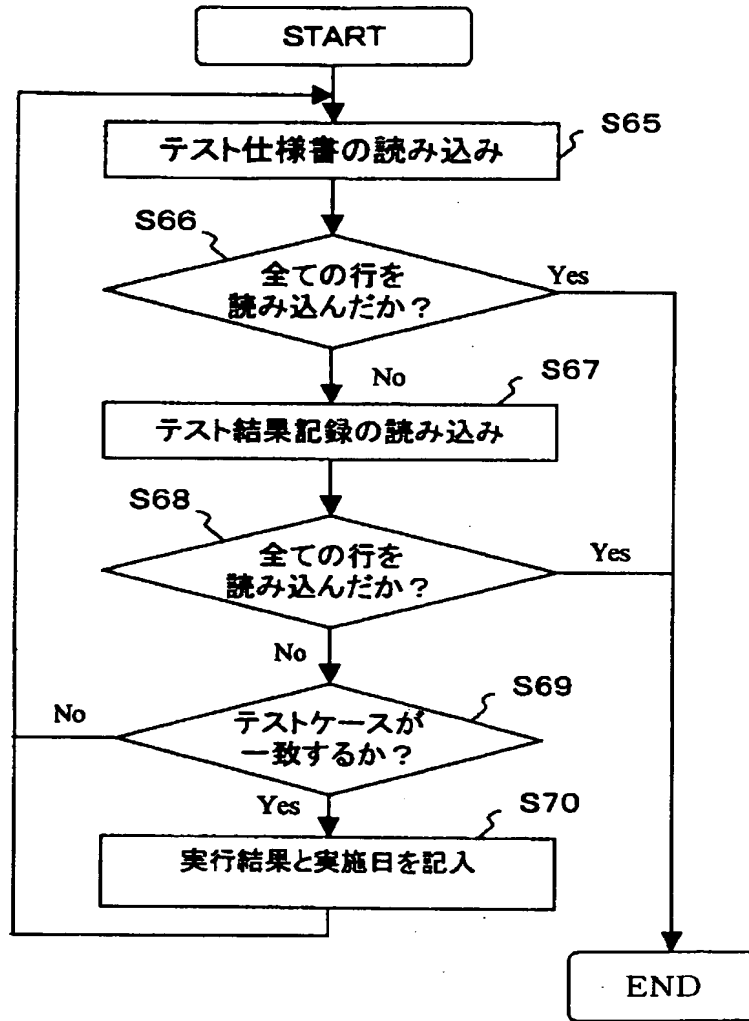
【図 3 1】

# 監視項目変化時の 性能測定支援装置の動作例の説明図



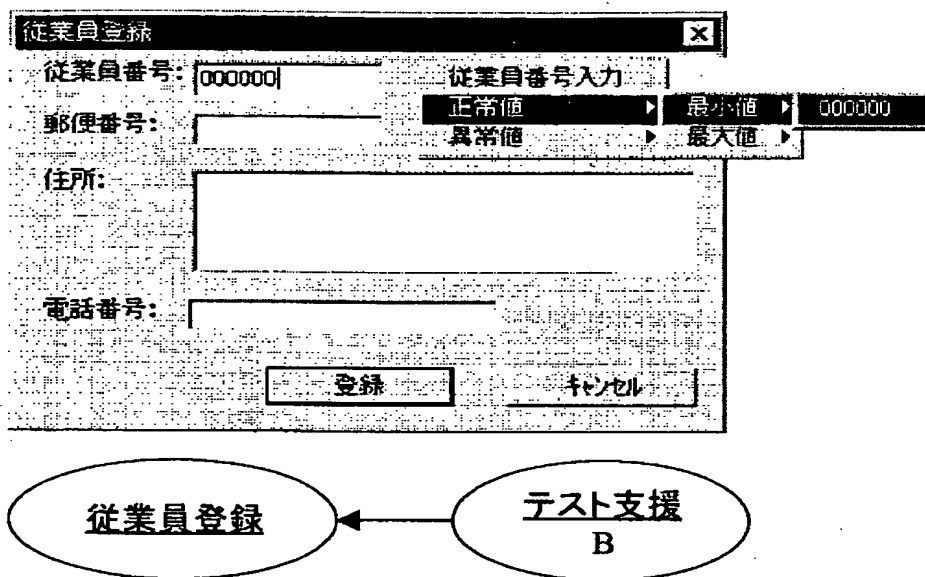
【図 3 2】

テスト報告書生成装置の処理フローチャート



【図33】

テスト環境における  
テスト支援Bクラスによるテスト支援の説明図





【図34】

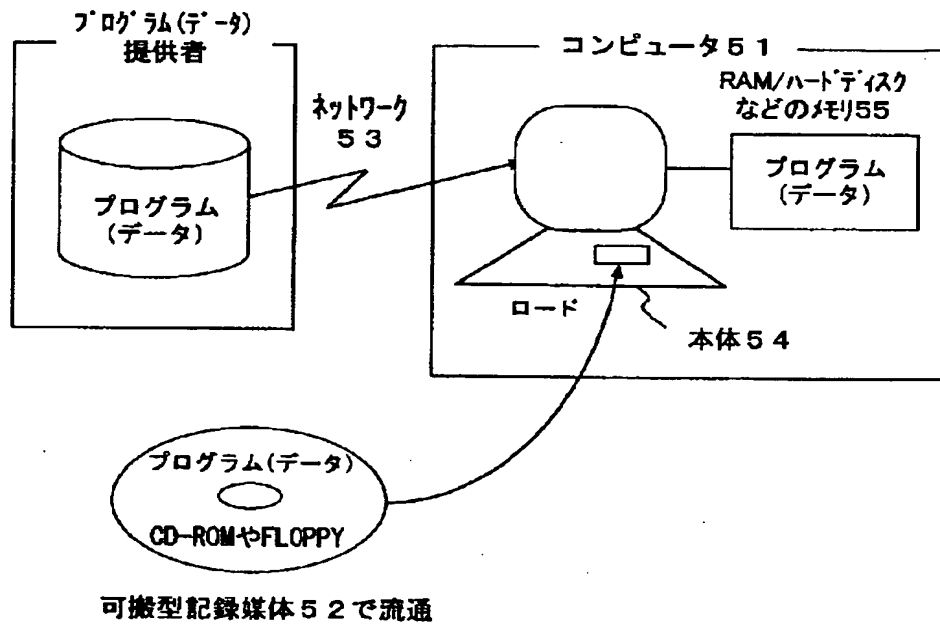
プログラム実行の本番環境における  
従業員登録画面を示す図

The screenshot shows a window titled "従業員登録" (Employee Registration). It contains several input fields: "従業員番号:" (Employee Number) with the value "000000", "郵便番号:" (Postal Code) with the value "1234567", "住所:" (Address) with the value "千葉市美浜区中瀬1-9-3", and "電話番号:" (Phone Number) with the value "0431234567". At the bottom right, there are two buttons: "登録" (Register) and "キャンセル" (Cancel).

従業員登録

【図 3 5】

本発明の実施形態を実現するためのプログラムの  
コンピュータへのローディングを説明する図



【書類名】 要約書

【要約】

【課題】 グラフィックユーザインタフェースを使用する画面プログラムのテストを自動化、効率化する。

【解決手段】 テスト対象画面プログラムの画面定義情報を入力として受け取り、該テスト対象画面プログラムのクラスに対してオブジェクト指向の継承の関係におけるサブクラスであって、画面プログラムをテストするためのテスト支援クラスを生成する手段 2 と、該生成されたテスト支援クラスを使用して、テスト対象画面プログラムのテストを実行する手段 3 とを備える。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [000005223]

1. 変更年月日 1996年 3月26日  
[変更理由] 住所変更  
住 所 神奈川県川崎市中原区上小田中4丁目1番1号  
氏 名 富士通株式会社